FONDUL SOCIAL EUROPEAN
Investeşte în oameni!
Programul Operaţional Sectorial pentru Dezvoltarea Resurselor Umane 2007 – 2013
Proiect POSDRU/6/1.5/S/19 – *Pregătirea competitivă a doctoranzilor în domenii prioritare ale societăţii bazate pe cunoaştere*

UNIVERSITATEA **POLITEHNICA** DIN BUCUREŞTI

Facultatea de Automatică şi Calculatoare

Catedra de Calculatoare

Nr. Decizie Senat 211 din 15.09.2011

TEZĂ DE DOCTORAT

Soluţii de asigurare a calităţii sistemelor software de monitorizare şi control a staţiilor electrice

Quality assurance solutions for electrical substations monitoring and control software systems

Autor: Ing. Victor Ursianu

COMISIA DE DOCTORAT

Preşedinte	Prof. Dr. Ing. Dumitru Popescu	de la	Facultatea de Automatică şi Calculatoare, Universitatea POLITEHNICA din Bucureşti
Conducător de doctorat	Prof. Dr. Ing. Florica Moldoveanu	de la	Facultatea de Automatică şi Calculatoare, Universitatea POLITEHNICA din Bucureşti
Referent	Prof. Dr. Ing. Sergiu Stelian Iliescu	de la	Facultatea de Automatică şi Calculatoare, Universitatea POLITEHNICA din Bucureşti
Referent	Prof. Dr. Ion Smeureanu	de la	Facultatea de Cibernetică, Statistică şi Informatică Economică, Academia de Studii Economice din Bucureşti
Referent	Prof. Dr. Ing. Mat. Dumitru Dan Burdescu	de la	Facultatea de Automatică, Calculatoare şi Electronică, Universitatea din Craiova

Bucureşti, 2011

Rezumat

Există mai multe puncte de vedere privind calitatea software, de la atribute de calitate internă, până la satisfacţia utilizatorului. În funcţie de scopul software-ului şi mediul în care funcţionează, anumiţi factori de calitate sunt mai importanţi decât alţii. Această teză încearcă să definească soluţiile adecvate pentru asigurarea calităţii unui sistem software de monitorizare şi control în domeniul energetic, sistem care poate fi încadrat în categoria sistemelor critice. Un astfel de sistem poate fi considerat o componentă dintr-un Smart Grid, un domeniu care capătă o din ce în ce mai mare amploare la nivel mondial şi care reprezintă viitorul în sectorul energetic.

Am încercat să clarific şi să propun soluţii de asigurare a calităţii sistemelor software de monitorizare şi control din domeniul energetic.

Sunt subliniate aspecte ale sistemelor software de monitorizare şi control precum şi al tehnicilor de asigurare a calităţii acestora.

Pentru testarea componentelor unui astfel de sistem, am dezvoltat un simulator pentru echipamentele de monitorizare tip IED (Intelligent Electronic Device), care poate fi configurat pentru diferite tipuri de echipamente electrice monitorizate. Acest simulator generează automat date de test, pornind de la domeniile de valori ale parametrilor echipamentelor electrice monitorizate.

Datorită faptului că standardul IEC61850 este unanim acceptat de către marii fabricanţi de echipamente de monitorizare şi control a staţiilor electrice, s-a utilizat limbajul SCL de configurare a unei staţii electrice pentru modelarea sistemului analizat în studiul de caz.

Pentru estimarea fiabilităţii unui sistem de monitorizare şi control a unei staţii electrice, am propus utilizarea modelului matematic al reţelei Bayesiene. De asemenea, am utilizat modelul Rayleigh pentru estimarea ratei de defectare a componentelor de monitorizare din cadrul sistemului din studiul de caz. Pentru estimarea momentului de timp al defectării sistemului am propus un model de tip lanţ Markov cu patru stări ale sistemului. Identificarea celor doua stări intermediare, între cea de funcţionare perfectă şi cea de nefuncţionare, poate aduce avantaje financiare imense, deoarece poate preveni ajungerea sistemului în starea de nefuncţionare, cu efecte imprevizibile (posibil catastrofale) datorită nefuncţionării staţiei electrice.

Aplicaţia software de modelare şi evaluare a reţelei Bayesiene, pe care am dezvoltat-o, permite preluarea configuraţiei staţiei electrice dintr-un fişier scris în limbajul SCL, completarea acestei configuraţii cu noi noduri şi calcularea probabilităţii de defectare a sistemului ţinând cont de probabilităţile de defectare ale fiecărui nod al reţelei.

Cuprins

Lista de figuri şi tabele

Lista de figuri

Lista de tabele

1. Introducere

1.1. Problema calităţii software

Problema calităţii sistemelor software de monitorizare şi control din domeniul energetic este de mare complexitate.

Calitatea sistemului depinde de calitatea tuturor componentelor sale, hardware şi software. Corectitudinea funcţionării unui sistem de timp real depinde atât de rezultatele calculelor cât şi de momentul în care sunt ele disponibile. În cazul sistemelor embedded, calitatea depinde de întregul ansamblu calculator-echipament.

În ultimii 10-15 ani, sistemele de timp real au evoluat ca urmare a cercetărilor şi rezultatelor deosebite din acest domeniu, care au fost imperativ cerute de necesitatea proiectării unor aplicaţii concrete, foarte complexe, critice, ce implică siguranţă şi predictibilitate deosebite: sisteme de control al traficului aerian, sisteme informaţionale distribuite, centrale nucleare, sisteme de apărare spaţiale, largi sisteme de comandă şi control în producţie, etc.

Aplicaţiile în timp real din prima generaţie rulau pe sisteme monoprocesor, problemele ce le rezolvau fiind relativ simple şi nepresupunând algoritmi sofisticaţi sau prelucrări foarte complexe. În domeniul energetic, aplicaţiile constau din conducerea centralelor electrice astfel încât să se reducă preţul de cost al energiei.

Pe lângă activitatea de control a instalaţiilor se cere şi monitorizarea producerii, transportului şi distribuirii. Timpii de răspuns sunt de ordinul microsecundelor sau milisecundelor pentru partea de control, ajungând până la o secundă pentru monitorizare. Numărul de sarcini (taskuri) este mare (de ordinul zecilor). Sistemele de acest tip au o mare funcţionalitate, iar complexitatea proiectării este ridicată. În cadrul interfeţei cu operatorii, se folosesc de obicei reprezentări grafice ale echipamentelor electrice monitorizate şi/sau controlate şi valorile numerice ale parametrilor achiziţionaţi pentru acestea.

În ceea ce priveşte calitatea software pentru aceste sisteme, vom folosi următorii termeni:

- **Defect (Defect):** o problemă software legată de comportarea sa externă sau caracteristicile sale interne, o anomalie în produsul software (“bug”).
- **Cădere (Failure**): incapacitatea unui sistem sau componentă de a-şi realiza funcțiile cerute conform specificațiilor de performanţă (IEEE 610.12, 1990).

- **Greșeală** (**Fault**): un pas, proces incorect, o definiție de date incorectă într-un program (IEEE 610.12).
- **Eroare** (Error): o acțiune umană care produce un rezultat incorect (IEEE 610.12).

Astfel, căderea poate fi interpretată ca o abatere comportamentală de la cerințele utilizatorului sau de la specificația produsului în timpul operării, greșeala ca fiind condiție existentă în software care cauzează o cădere iar eroarea reprezintă o acțiune umană absentă sau incorectă care are ca efect injectarea unor greșeli în produsul software.

1.2. Importanţa asigurării calităţii software

Datorită experienţei dobândite în domeniul calităţii software şi în ultimii ani în domeniul monitorizării echipamentelor electrice primare şi apoi al staţiilor electrice în totalitate, din punct de vedere al aparatajului primar, consider că aceste sisteme de monitorizare şi control pot fi îmbunătăţite. Există mai multe aspecte ce pot fi studiate, inclusiv pe partea de proiectare, dezvoltare şi mentenanţă a acestor sisteme.

Mai întâi de toate, trebuie conştientizată importanţa domeniului energetic. Acesta este un domeniu fundamental al dezvoltării economice naţionale şi internaţionale. Fără energie, nu se poate dezvolta nimic. Pentru dezvoltarea conceptului de Smart Grid (Reţea Inteligentă) sunt puse la dispoziţie fonduri financiare uriaşe la nivelul Uniunii Europene şi nu numai.

Există o părere unanimă a specialiştilor în acest domeniu că domeniul energetic şi viitorul Smart Grid nu poate funcţiona fără monitorizarea parametrilor echipamentelor electrice din cadrul staţiilor electrice.

Staţiile electrice reprezintă zona intermediară cea mai importantă între producătorul de energie (hidrocentrale, termocentrale, centrale nucleare, eoliene, etc.) şi utilizatorul final.

În funcţie de tipul lor constructiv, staţiile electrice se împart în mai multe categorii. În cadrul tezei de doctorat, m-am referit la staţiile electrice de transformare aparţinând reţelei electrice naţionale de transport a energiei electrice (RET). Soluţiile de asigurare a calităţii software prezentate, împreună cu modelele matematice propuse pot fi folosite fără probleme şi la alte tipuri de staţii electrice.

Calitatea unui produs software este dată de “capacitatea sa de a putea fi utilizat eficient, efectiv şi confortabil, de către un set de utilizatori, pentru un set de scopuri, în condiţii specificate“.

Asigurarea calităţii software a sistemului este foarte importantă din necesitatea de a avea un produs fiabil, cu cât mai puţine defecte şi căderi. Personalul din staţie este

principalul beneficiar al unui produs fiabil întrucât foloseşte informaţiile furnizate de sistem pentru raportări către dispecerat şi mai apoi pentru a lua decizii operative. Totodată informaţiile achiziţionate de la sistemul de monitorizare şi control sunt raportate mai departe la dispeceratul zonal şi mai apoi la dispeceratul energetic naţional (DEN), principalul organism din domeniu care are date despre situaţia tuturor staţiilor electrice la orice moment de timp. Toate deciziile de retragere din exploatare sau comenzile importante pentru echipamentele electrice se iau cu aprobarea sau la indicaţiile dispeceratului. De aceea, orice informaţie eronată transmisă sau orice întârziere de raportare a unui defect sau a unei avarii poate duce la funcţionarea imperfectă şi la decizii greşite luate de către dispecerat.

În cazul cel mai rău, situaţia poate genera un dezechilibru energetic în zona respectivă şi pagube materiale importante pentru consumatori.

1.3. Structura şi obiectivele tezei

Următoarele capitole ale tezei sunt:

- Capitolul 2, intitulat "Asigurarea calităţii software" descrie pe scurt standardul ISO9126 şi modelul de calitate software împărţit în 6 caracteristici de calitate şi 21 de subcaracteristici. Sunt descrise categoriile în care se împart metricile software: metrici de dimensiune, complexitate, calitate a produsului şi respectiv metrici interne ale procesului.

  Sunt prezentate soluţii de asigurare a calităţii software, care presupun utilizarea de tehnici şi metode de prevenire a injectării defectelor precum şi tehnici de eliminare a defectelor şi izolare a acestora.

  S-a pus accentul pe activităţile de inspecţie şi testare, utilizarea blocurilor de recuperare, N-version programming, self-checking precum şi tehnici mai noi ca reconfigurare şi reîntinerire, BASE. De asemenea, sunt descrise pe scurt aspecte de asigurare a siguranţei transmisiei datelor în cadrul unei staţii electrice.

- Capitolul 3, intitulat "Analiza aspectelor de calitate ale unui sistem software de monitorizare şi control a unei staţii electrice" prezintă arhitectura generală a unui astfel de sistem. Sunt enumerate sisteme existente produse de fabricanţi cunoscuţi în domeniul energetic.

Este prezentat sistemul de monitorizare şi control EMCSIT care a fost utilizat ca şi studiu de caz. Se face o analiză a aspectelor de calitate software plecând de la un punct de vedere general şi apoi sunt descrise particularităţile unui sistem software de monitorizare şi control a unei staţii electrice. Sunt identificate problemele care pot apărea în dezvoltarea, funcţionarea şi operarea acestui sistem.

Pornind de la rezultatele acestor analize, sunt propuse tehnici de asigurare a calităţii software pentru sistemele de monitorizare şi control a unei staţii electrice şi estimarea fiabilităţii acestora.

- Capitolul 4, intitulat "Studiu de caz: asigurarea calităţii software pentru sistemul de monitorizare şi control EMCSIT" prezintă arhitectura şi funcţionalitatea sistemului EMCSIT de monitorizare şi control a unei staţii electrice din România.

  Sunt descrise soluţiile şi exemple de aplicare a acestor soluţii pentru asigurarea şi îmbunătăţirea calităţii software a acestui sistem. Pentru îmbunătăţirea procesului de testare a IED-urilor incluse în acest sistem este propusă dezvoltarea şi utilizarea unui simulator software ce generează pachete de date şi le trimite pe interfaţa serială a calculatorului server. Aceste date sunt achiziţionate de aplicaţiile software tip server EMCSIT dezvoltate pentru fiecare tip de IED şi care sunt instalate pe acelaşi calculator. Datele de intrare pentru simulator sunt generate utilizând limitele tehnologice pentru mărimile electrice monitorizate.

  Aplicând metoda *Pairwise testing* se generează mult mai puţine cazuri de test faţă de numărul total de combinaţii însă acestea pot conduce la descoperirea a aproximativ 70% din erori în timpul testării.

- Capitolul 5, intitulat "Standardul IEC61850" descrie limbajul SCL de configurare a unei staţii electrice şi aspecte generale ale standardului IEC61850. Acest standard a fost ales întrucât este unanim acceptat de producătorii de echipamente de monitorizare şi control din domeniu.

  Sunt prezentate aspecte generale privind IEC61850 şi utilizarea acestuia în cadrul unei staţii electrice. Este prevăzută folosirea IED-uri (Intelligent Electronic Device) pentru monitorizarea echipamentelor electrice primare. IED-ul reprezintă, conform IEC61850 orice echipament ce include unul sau mai multe procesoare

(microcontrollere) cu posibilitatea de a primi sau trimite date de la sau către o sursă externă, sau de a controla acea sursă.

Limbajul SCL a fost ales în continuare pentru configurarea unei staţii electrice ce include componentele unui sistem software de monitorizare şi control. Structura generală a unui fişier scris în limbajul SCL permite descrierea echipamentelor electrice din cadrul unei staţii precum şi asocierea IED-urilor destinate monitorizării şi respectiv controlului acestora.

- Capitolul 6, intitulat " Utilizarea unor modele matematice pentru estimarea fiabilității sistemelor de monitorizare și control a stațiilor electrice" prezintă modelul de distribuţie Rayleigh şi rezultatele aplicării acestui model asupra ratelor de defectare obţinute pentru sistemul EMCSIT folosit ca studiu de caz.

  Este prezentat modelul matematic al lanţurilor Markov şi o propunere de implementare a acestui model. Se propune încadrarea unei aplicaţii software în 4 stări posibile: stare bună, acceptabilă, proastă respectiv inacceptabilă. Sunt propuse mai multe criterii pentru a încadra aplicaţia software într-una din stări, cel ales de mine se bazează pe rata pachetelor de date neachiziţionate de la server. În urma modelării aplicaţiei software utilizând un lanţ Markov, se poate estima starea curentă a sistemului şi timpul de trecere dintr-o stare în alta. Se poate prezice momentul când acest sistem va cădea sau se va defecta.

  Tot în acest capitol, se prezintă modelarea unei reţele Bayesiene având drept noduri tip părinte, IED-urile ce monitorizează echipamentele electrice. Cu ajutorul informaţiilor privind rata de defectare/cădere a acestora, se propune estimarea fiabilităţii sistemului de monitorizare şi control prin calculul probabilităţii de defectare a nodurilor reţelei şi a întregului sistem.

- Capitolul 7, intitulat " Aplicaţie software pentru estimarea fiabilității unui sistem de monitorizare şi control a unei staţii electrice" descrie componentele incluse în aplicația dezvoltată pentru estimarea fiabilităţii sistemului: *Parser SCL* şi *Calcul BN*. Pornind de la configuraţia unei staţii electrice descrisă în limbajul SCL, sunt colectate informaţii privind IED-urile ce monitorizează echipamentele electrice. Modulul *Parser SCL* citeşte informaţiile din fişierul scris în limbajul SCL şi pune la dispoziţia modulului *Calcul BN*, date reprezentând IED-urile instalate în staţie. Acestea

reprezintă nodurile reţelei Bayesiene. Se pot edita legăturile între noduri, definind cele două tipuri de noduri: nod de tip fiu şi nodurile de tip părinte.

Putem completa configuraţia staţiei electrice în limbajul SCL conform standardului IEC61850, adăugând noduri de tip calculator Server- local, noduri de tip calculator Client, noduri de tip Server de baze de date, etc. Utilizând informaţii privind rata de defectare/cădere a IED-urilor, modulul *Calcul BN* poate calcula probabilitatea de defectare a întregului sistem de monitorizare şi control a staţiei electrice.

- Capitolul 8, intitulat "Concluzii, contribuţii proprii şi planuri de cercetare pentru viitor" prezintă concluziile cercetării şi contribuţiile proprii originale dezvoltate şi aplicate în scopul asigurării calităţii sistemelor software de monitorizare şi control a staţiilor electrice. Sunt propuse direcţii de cercetare pentru viitor.

Obiectivele tezei sunt următoarele:

- Identificarea modalităţilor de îmbunătăţire a procesului de dezvoltare software pentru asigurarea unui nivel ridicat de fiabilitate şi a unei mentenabilităţi crescute pentru un sistem de monitorizare şi control a unei staţii electrice.
- Crearea unor instrumente software de testare a modulelor sistemului şi optimizarea acestora.
- Alegerea unor modele matematice utile pentru estimarea fiabilităţii la nivel de componentă precum şi la nivelul întregului sistem.
- Dezvoltarea unor module software pentru estimarea fiabilităţii sistemului.

2. Asigurarea calităţii software

2.1. Calităţile generale ale unui produs software

O metrică software este o măsură a unei proprietăţi pentru un artefact software. O metrică software trebuie să fie cuantificabilă şi să poată fi aplicată unei caracteristici a produsului software. De exemplu, mentenabilitatea (uşurinţa de întreţinere), este o caracteristică a produsului software şi este precizată în toate modelele principale de asigurare a calităţii. O metrică software semnificativă pentru această caracteristică este timpul mediu de reparare a unui defect. O măsură mai bună a cauzei mentenabilităţii scăzute a unui produs software poate fi şi complexitatea codului. Pentru aceasta a fost dezvoltată o metrică de către Thomas McCabe încă din anul 1976. Această metrică permite evaluarea cantitativă a oricărei secvenţe de cod sursă. Complexitatea codului poate fi evaluată static, dar timpul mediu de rezolvare a unui bug poate fi estimat în timpul testelor de sistem sau în timpul exploatării produsului.

Există două puncte de vedere asupra calităţii software: un punct de vedere extern (A) şi respectiv unul intern (B).

A) Calităţile software interne sunt strâns legate de metodele de proiectare şi implementare. Ele nu sunt vizibile clientului/utilizatorului dar influenţează puternic calităţile externe, mai ales fiabilitatea, eficienţa, portabilitatea şi uşurinţa de întreţinere.

Câteva dintre acestea sunt:

- coeziunea strânsă la nivelul fiecărui modul (funcţie, clasă);
- cuplare scăzută între module;
- complexitate redusă a proiectării şi la nivelul codului;
- claritatea codului şi a documentării sale;

B) Calitatea externă înseamnă toate proprietăţile software-ului pe care utilizatorii săi le pot percepe şi prin care îl pot aprecia:

- conformitatea cu aşteptările lor (şi evoluţia acestora);
- fiabilitatea;
- precizia/acurateţea;
- uşurinţa de utilizare şi confortul (inclusiv întârzierea răspunsului);
- robusteţea (sau adaptabilitatea la condiţii neprevăzute de utilizare);
- adaptabilitate pentru extinderi viitoare sau evoluţii;

O aplicaţie software care prezintă calitate internă ridicată este uşor de modificat, uşor de extins cu noi facilităţi şi uşor de testat. Software-ul cu o calitate internă scăzută este greu de înţeles, greu de schimbat şi dificil de extins. Măsuri cum ar fi cele definite de McCabe (complexitate ciclomatică, coeziune, cuplare), punctele funcţiune (introdusă de Allan Albrecht de la IBM) pot fi folosite pentru a estima calitatea internă.

Calitatea software externă este o măsură a modului în care sistemul în ansamblul său îndeplineşte cerinţele beneficiarului. Aceasta ne ajută să răspundem la următoarele întrebări: Sistemul asigură funcţionalitatea necesară? Este interfaţa clară şi coerentă? Software-ul produce valoarea aşteptată a afacerii?

Încercările de standardizare a terminologiei referitoare la calitatea produselor software au condus la standardul ISO 9126 (Information Technology - Software Product Quality, Part 1: Quality Model, 1998).

Sunt definite 6 caracteristici de calitate, împărţite în 21 de subcaracteristici.

1) **Funcţionalitatea**: realizarea scopului de bază pentru care a fost realizat produsul.
   - **Oportunitatea:** prezenţa unui set de funcţii adecvate pentru taskuri specificate;
   - **Precizia:** furnizarea unor rezultate sau efecte corecte sau agreate;
   - **Interoperabilitatea:** capacitatea produsului de a interacţiona cu sisteme specificate;
   - **Securitatea**: capacitatea de a preveni accesul neautorizat, accidental sau deliberat, la programe sau date;
   - **Conformitatea:** adeziunea la standarde, convenţii, legi şi protocoale.

2) **Fiabilitatea:** capacitatea produsului de a-şi menţine nivelul de performanţă, în condiţii definite, pentru o perioadă de timp definită.
   - **Maturitatea:** atribut bazat pe frecvenţa căderilor datorate greşelilor în software;
   - **Toleranţa la defecte (robusteţea):** capacitatea de a-şi menţine un nivel de performanţă specificat în cazuri de căderi software sau intrări neaşteptate;
   - **Restabilirea după căderi:** capacitatea şi efortul necesar pentru restabilirea nivelului de performanţă, recuperarea datelor afectate, după posibile căderi;
   - **Conformitatea:** adeziunea la standarde, convenţii, legi şi protocoale.

3) **Utilizabilitatea (uşurinţa de utilizare)**: efortul necesar pentru utilizarea sa de către un set de utilizatori definit.

- **Uşurinţa de înţelegere:** efortul solicitat unui utilizator de a recunoaşte conceptul logic şi aplicabilitatea sa;
- **Uşurinţa de învăţare:** efortul solicitat unui utilizator de a învăţa aplicaţia, operarea, intrările şi ieşirile;
- **Operabilitatea:** uşurinţa de operare şi de control de către utilizatori;
- **Puterea de atracţie:** capacitatea produsului de a fi atrăgător pentru utilizatori;
- **Conformitatea:** adeziunea la standarde, convenţii, legi şi protocoale.

4) **Eficienta:** relaţia între nivelul de performanţă al produsului şi cantitatea de resurse utilizate, în condiţii definite.
   - **Timp la execuţie:** viteza de răspuns, timpi de prelucrare, rata ieşirilor la realizarea funcţiilor;
   - **Utilizarea resurselor:** cantitatea de resurse utilizate şi durata utilizării pentru realizarea funcţiilor sale;
   - **Conformitatea:** adeziunea la standarde, convenţii, legi şi protocoale.

5) **Uşurinţa de întreţinere**: efortul necesar pentru efectuarea modificărilor, inclusiv corecţii, îmbunătăţiri sau adaptări ale produsului la schimbări ale mediului de funcţionare, a cerinţelor şi schimbărilor funcţionale.
   - **Uşurinţa de analiză:** efortul necesar pentru diagnoza defectelor, a cauzelor căderilor, pentru identificarea părţilor care trebuie să fie modificate;
   - **Uşurinţa de modificare:** efortul necesar pentru înlăturarea defectelor sau pentru schimbări;
   - **Stabilitatea:** riscul efectelor neaşteptate în urma modificărilor;
   - **Uşurinţa de testare:** efortul necesar pentru a valida produsul modificat;
   - **Conformitatea:** adeziunea la standarde, convenţii, legi şi protocoale.

6) **Portabilitatea:** capacitatea produsului de a fi transferat de la o organizaţie sau platformă software/hardware la o alta.
   - **Adaptabilitatea:** capacitatea de adaptare la diferite medii specificate;
   - **Uşurinţa de instalare:** efortul necesar pentru instalarea produsului într-un mediu specificat;

- **Co-existenţa:** capacitatea de a co-exista cu alte produse independente în acelaşi mediu;
- **Oportunitatea** şi efortul necesar pentru a folosi produsul în locul altui produs într-un mediu particular;
- **Conformitatea:** adeziunea la standarde, convenţii, legi şi protocoale.

2.2. Metrici pentru evaluarea calităţii software

O metrică software este o măsură a unei proprietăţi a unui artefact software. O metrică software trebuie să fie cuantificabilă şi să poată fi aplicată unei caracteristici a produsului software.

2.2.1. Categorii de metrici software

Sunt următoarele categorii de metrici:

a. Metrici de dimensiune (size metrics)

Metricile de dimensiune sunt folosite pentru aprecierea aspectelor legate de calitate, productivitate şi de estimare a costurilor.

- **Linii de cod (LOC, Lines Of Code)**

Există în următoarele variante:

- LOC : Lines of Code (linii de cod);
- KLOC : Kilo Lines of Code (kilo-linii de cod);

Este o metrică simplă dacă se specifică ce înseamnă "codul". Pentru aceasta există următoare opţiuni:

- Linii de cod executabile;
- Linii de cod executabile, definiţii de date şi comentarii;
- Linii de cod fizice;
- Altele;

Sunt disponibile următoarele metrici, mai precise faţă de LOC şi respectiv KLOC:

- DSI: instrucţiuni sursă livrate;
- KDSI: kilo-instrucţiuni sursă livrate;
- SSI: instrucţiuni sursă expediate;
- CSI: instrucţiuni sursă noi şi schimbate;

SSI (lansarea curentă) are următoarea formulă de calcul: SSI (lansarea precedentă) + CSI (pentru versiunea curentă) – cod şters – cod schimbat (pentru a evita numărarea repetată în SSI şi CSI).

- **Puncte funcţiune (FP - function points)**

Punctele funcţiune reprezintă dimensiunea funcţională a unui sistem. Analiza punctelor funcţiune (FPA, Function Point Analysis) este o metodă certificată ISO pentru măsurarea dimensiunii funcţionale a unui sistem informatic. Dimensiunea funcţională reflectă gradul de funcţionalitate care este recunoscut şi relevant pentru utilizator şi este independentă de tehnologia folosită pentru implementarea sistemului.

Indiferent de limbajul de programare sau mediul în care rulează sistemul, numărul de puncte funcţiune pentru un sistem va fi acelaşi. [Swq9-15].

Dimensiunea funcţională, exprimată în puncte funcţiune, poate fi folosită pentru:

- Estimarea bugetului pentru dezvoltarea aplicaţiei sau costurile extinderii;
- Estimarea bugetului anual pentru costurile de mentenanţă la portofoliul de aplicaţii;
- Determinarea productivităţii proiectului după încheierea lui;
- Determinarea dimensiunii software-ului pentru estimarea costurilor;

FPA (analiza punctelor funcţiune) poate fi folosită şi pentru estimarea efortului de testare necesar pentru dezvoltarea software.

Aceasta metrică a fost folosită pentru prima dată în cadrul companiei IBM (Albrecht [Swq9-16]) la mijlocul anilor 1970. Este o metrică empirică, bazată pe studierea proiectelor dezvoltate în această companie.

Metrica punctelor funcţiune este considerată mai stabilă şi mai relevantă faţă de metricile bazate pe LOC (lines of codes).

b. Metrici de complexitate

Complexitatea ciclomatică utilizează graful de control al programului. Ea porneşte de la ipoteza că dificultatea de înţelegere a unui program este în mare măsură determinată de complexitatea grafului. Este o măsură a dificultăţii de testare a modulelor şi a fiabilităţii la nivel de modul.

Complexitatea unei entităţi este determinată de relaţiile dintre părţile sale. Părţile unui modul software sunt instrucţiunile sale. Relaţiile dintre ele se bazează pe trei structuri:

- **Secvenţa**: bloc maximal indivizibil de instrucţiuni care se execută întotdeauna în aceeaşi ordine;
- **Selecţia**: condiţia;
- **Iteraţia**: ciclul. Iteraţia poate fi simulată prin salt la o condiţie.

Complexitatea ciclomatică este definită pentru un modul pe baza grafului de control al modulului. McCabe defineşte complexitatea ciclomatică a unui graf de control astfel:

$$v = e - n + 2$$

unde:

- e este numărul de arce (edges);
- n este numărul de noduri.

Pentru o secvenţă: v=1; este necesară o singură execuţie de test pentru a executa fiecare instrucţiune din secvenţă.

Fiecare salt adăugat la un modul creşte complexitatea sa cu 1 şi necesită o execuţie de test suplimentară pentru testarea sa.

Deci, complexitatea ciclomatică a unui modul dă numărul minim de execuţii de test ale unui modul, pentru acoperirea tuturor arcelor.

Complexitatea ciclomatică totală a unui program se obţine însumând complexităţile ciclomatice ale modulelor sale. Este exprimată prin formula:

$v = e - n + 2p$, unde:

- p este numărul de module;
- e este numărul de arce din toate modulele;
- n este numărul de noduri din toate modulele;

O formulă echivalentă este:

$$v = v_1+v_2+..+v_p$$

unde v_1, v_2, …, v_p sunt complexităţile ciclomatice ale celor p module.

Combinând mai multe module într-unul singur, va rezulta un modul cu o complexitate ciclomatică mai mare decât a modulelor combinate dar mai mică în comparaţie cu complexitatea totală.

Descompunerea unui modul în module mai mici creşte complexitatea totală dar reduce complexitatea la nivelul fiecărui modul.

Complexitatea proiectării unui modul, notată de McCabe cu iv, măsoară efectul individual al unui modul asupra proiectului general pentru un program. Complexitatea proiectării unui modul este evaluată plecând de la graful de control al modulului şi marcând nodurile care conţin apeluri de module externe.

Graful de control este apoi redus după următoarele reguli:

1. nodurile marcate nu pot fi eliminate;
2. nodurile nemarcate care nu conţin decizii sunt eliminate;
3. arcele care întorc controlul la începutul unui ciclu care conţine numai noduri nemarcate sunt eliminate;
4. arcele care unesc nodul de start al unei instrucţiuni *case* cu nodul de sfârşit sunt eliminate dacă nici una dintre celelalte ramificaţii ale instrucţiunii *case* nu conţin noduri marcate.

Complexitatea proiectării unui ansamblu de module are următoarea formulă:

$S_0 = iv_1 + iv_2 .. + iv_n$,

unde iv_1, iv_2, iv_n sunt complexităţile de proiectare ale modulelor din ansamblu.

Complexitatea integrării unui ansamblu de N module, notată de McCabe cu S1, este definită astfel:

$S_1 = S_0 - N + 1$

Complexitatea integrării a N module care nu conţin ramificaţii este deci 1.

Formal, complexitatea integrării necesită măsurarea complexităţii proiectării fiecărui modul.

În timpul proiectării arhitecturale, de regulă, nu este disponibil graful fluxului de control al fiecărui modul. Totuşi, ar trebui să fie disponibilă suficientă informaţie pentru a defini complexitatea proiectării fiecărui modul, chiar fără a cunoaşte logica modulelor. Diagrama de structură, care reflectă apelurile modulelor programului, poate furniza aceste informaţii.

c. Metrici de calitate a produsului

Calitatea produsului reprezintă "totalitatea caracteristicilor ce îi conferă acestuia aptitudinea de a satisface nevoile utilizatorului" [Swq9-14].

Calitatea din punctul de vedere al producătorului reprezintă conformitatea cu cerinţele. Din punctul de vedere al utilizatorului, calitatea reprezintă conformitatea cu aşteptările utilizatorului.

Câteva atribute externe nu pot fi măsurate: uşurinţa de utilizare, uşurinţa de întreţinere, uşurinţa de instalare.

Sunt două nivele ale metricilor de calitate a produsului software :

- Metrici de calitate intrinsecă;
- Metrici orientate către client;

Metrici de calitate intrinsecă a produsului

Astfel de metrici, care referă calităţile externe ale unui produs software sunt fiabilitatea şi densitatea defectelor. Aceste metrici sunt corelate dar sunt diferite. Ambele au valori bazate pe predicţii.

Fiabilitatea software reprezintă probabilitatea ca un program să își îndeplinească funcţiile specificate, pentru o perioadă de timp definită, în condiţii specificate. De obicei, fiabilitatea este estimată în timpul testelor de sistem, utilizând teste statistice, bazate pe profilul utilizării software-ului.

Metricile de fiabilitate sunt următoarele:

- MTBF (timpul mediu între defectări); este timpul aşteptat între două căderi succesive ale sistemului, exprimat în ore. Este o metrică importantă de fiabilitate pentru sistemele ce pot fi reparate sau restaurate (sisteme reparabile).
- MTTF (timpul mediu până la defectare) reprezintă timpul până la căderea sistemului. În termeni de fiabilitate, este o metrică pentru sisteme nereparabile. Sistemele nereparabile pot cădea doar o singură dată.
- Timpul mediu de reparare (MTTR) reprezintă timpul mediu pentru repararea unui sistem după o cădere.

Atunci când nu sunt întârzieri în procesul de reparare (de exemplu, datorită unor întârzieri de aprovizionare cu componente hardware) : MTBF = MTTF + MTTR

Sistemele software sunt sisteme reparabile. Modelele de fiabilitate software neglijează timpul necesar pentru repararea sistemului după o cădere.

Dacă MTTR =0 atunci MTBF = MTTF

Disponibilitatea unui sistem are următoarea formulă: MTTF / MTBF = MTTF / (MTTF + MTTR), iar indisponibilitatea este: 1 – Disponibilitatea.

Metrici orientate către client

Aceste metrici caracterizează problemele utilizatorului atunci când folosesc produsul software: defecte valide, probleme de utilizare, documentaţie neclară, erori în utilizare.

Una din metrici este PUM (probleme per utilizator pe lună):

PUM = TNP/ TNM, unde

- TNP reprezintă numărul total de probleme raportate de către clienţi pentru o anumită perioadă de timp.
- TNM reprezintă numărul total de licenţe instalate pentru software înmulţit cu numărul de luni de utilizare ale software-ului în timpul perioadei de testare.

Există şi o măsură a satisfacţiei utilizatorului, evaluată pe 5 nivele:

- Foarte satisfăcut: toţi utilizatorii sunt complet satisfăcuţi;
- Satisfăcut: toţi utilizatorii sunt satisfăcuţi sau 50% sunt complet satisfăcuţi şi 50% sunt indiferenţi;
- Indiferent: 50% din utilizatorii sunt satisfăcuţi;
- Nesatisfăcut 25% din utilizatorii sunt satisfăcuţi;
- Complet nesatisfăcut: toţi utilizatorii sunt complet nesatisfăcuţi;

IBM a dezvoltat modelul CUPRIMDSO pentru certificarea calităţii software care are următoarele componente:

- Capability – capabilitate/funcţionalitate;
- Usability – utilizabilitate;
- Performance – performanţă;
- Reliability – fiabilitate;
- Installability – instalabilitate;
- Mentenability – mentenabilitate;
- Documentation – documentare/informaţii;
- Service;
- Overall – calitatea per ansamblu.

Similar, Hewlett-Packard a dezvoltat modelul FURPS cu componentele:

- Functionability – funcţionalitate;

- Usability – utilizabilitate;
- Reliability – fiabilitate;
- Performance – performanţă;
- Service;

Alte metrici existente sunt procentul de clienţi complet satisfăcuţi, de clienţi satisfăcuţi, parţial nesatisfăcuţi respectiv nesatisfăcuţi.

d. Metrici interne ale procesului (in-process metrics)

Metricile interne ale procesului au rolul de a exprima starea software-ului din punct de vedere practic şi a susţine eliminarea defectelor înainte de livrarea aplicaţiei software către beneficiar.

Sunt disponibile următoarele metrici:

- Densitatea defectelor în timpul testării

Densitatea defectelor în timpul testării formale (teste de sistem) este estimată prin numărul de defecte per KLOC sau numărul de defecte per puncte funcţiune. Este de obicei corelată cu rata defectelor în timpul operării: cu cât sunt descoperite mai multe defecte în timpul testării, cu atât mai mult vor fi găsite defecte mai târziu. Această metrică este utilizată pentru a estima evoluţia produsului între două livrări.

- Modelul apariţiei defectelor în timpul testelor formale

Oferă mai multe informaţii faţă de densitatea defectelor: având aceeaşi rată a defectelor per ansamblu, diferite modele de apariţie a defectelor indică diverse nivele de calitate în timpul operării.

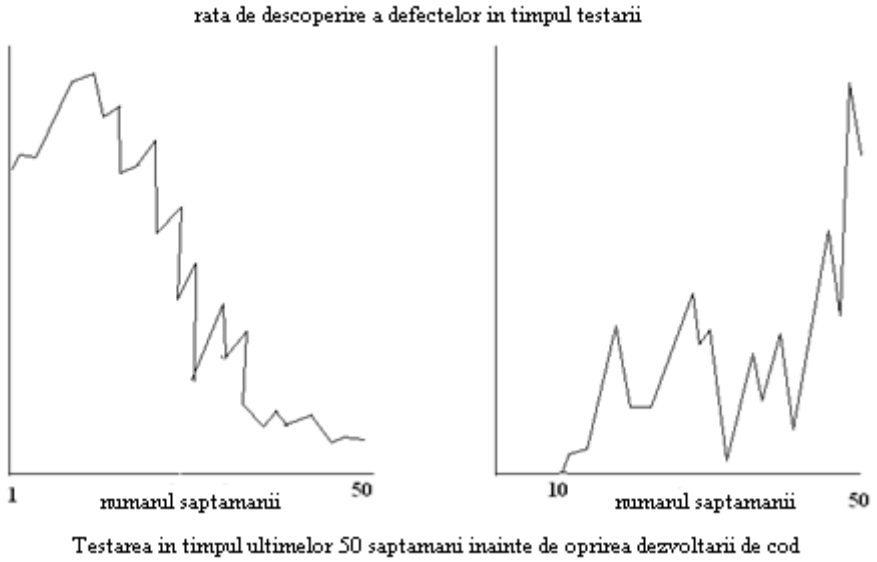

Fig. 2.2.1.1. Rata apariţiei defectelor în timpul testării

Unitatea de timp pentru observarea apariţiei defectelor este de obicei săptămâna, ocazional luna. Pentru modelele de fiabilitate ce au nevoie de timp de execuţie, unitatea de timp este timp CPU (procesor).

2.2.2. Metrici integrate în Visual Studio .NET

În continuare sunt prezentate câteva metrici software care pot fi calculate şi utilizate în cadrul proiectelor dezvoltate în mediul de programare Microsoft Visual Studio .NET:

- **Index de mentenabilitate (Maintainability Index);**

Indicele de mentenabilitate (MI) se bazează pe un set de valori numerice dezvoltat la Universitatea din Idaho, folosind formula Halstead pentru efort şi complexitatea ciclomatică McCabe, plus alţi factori referitori la numărul de linii de cod şi procentul de comentarii. MI este utilizat în principal pentru a determina gradul de dificultate al mentenanţei codului: înalt, mediu sau redus.

Acesta este independent de limbaj şi a fost validat în domeniu de către Hewlett-Packard (HP). HP a concluzionat că modulele cu un MI de cel puţin 65 sunt dificil de întreţinut, module între 65 şi 85 au întreţinere rezonabilă şi cele cu MI peste 85 au mentenabilitatea excelentă.

În cadrul mediului de dezvoltare Microsoft Visual Studio .NET 2010, indexul de mentenabilitate a fost calculat iniţial după cum urmează:

Index Mentenabilitate = 171 - 5,2 * ln(Volum Halstead) - 0,23 * (Complexitatea Ciclomatică) - 16,2 * ln(linii de cod)

Aceasta a însemnat că valoarea a variat de la 171 la un număr negativ nemărginit. S-a observat că în situaţiile când valoarea a tins către 0, a fost în mod clar greu să se facă mentenanţă pentru cod, atunci diferenţa între valoarea 0 şi o valoare negativă nu a fost folositoare. Ca urmare a utilităţii scăzute pentru numere negative şi dorinţei de a păstra această metrică cât mai clară, s-a decis ca toate valorile negative sau egale cu 0 să aibă valoarea 0 şi apoi au fost scalate valorile mai mici ca 171 pentru a fi în gama de la 0 la 100.

Astfel, formula folosită în prezent este:

Index Mentenabilitate= MAX(0, (171 - 5,2 x ln(Volumul Halstead) - 0,23 x (Complexitatea Ciclomatică) - 16,2 x ln(linii de cod)) x 100 / 171)

Volumul Halstead reprezintă volumul informaţiei (exprimată în biţi) necesar pentru a realiza un program software.

Volumul Halstead = N * log2(n) unde N este dimensiunea programului iar n reprezintă numărul operanzilor din cadrul programului.

Volumul Halstead descrie dimensiunea implementării unui algoritm. Calculul acestuia se bazează pe numărul de operaţii efectuate şi operanzii folosiţi în algoritm.

Deoarece indexul de mentenabilitate are valori între 0 şi 100, acest interval a fost împărţit în 0-9, 10-19 şi 20-100, pentru a semnaliza doar codul care a fost într-adevăr suspect.

Avem următoarea clasificare:

- 0-9 = nivel roşu;
- 10 - 19 = nivel galben;
- 20-100 = nivel verde;

Atunci când nivelul este roşu, putem spune cu un grad ridicat de încredere că există o problemă cu codul.

- **Complexitate ciclomatică (Cyclomatic Complexity);**

Cu cât sunt utilizate mai multe instrucţiuni: *if, switch, while do* şi altele similare, cu atât acest număr va fi mai mare. Trebuie menţinută o valoare mai mică de 20 pentru fiecare

metodă. O valoare mai mare indică faptul că acel cod este complicat şi greu de întreţinut. Este preferată o metodă cu o complexitate ciclomatică mică.

- **Adâncimea arborelui de moştenire al unei clase (Depth of Inheritance);**

Aceasta metrică reprezintă cel mai lung drum de la clasa de bază până la o clasă derivată din aceasta.

Cu cât o clasă se află mai jos în ierarhie, cu atât creşte probabilitatea să moştenească mai multe metode de la clasele pe care le moşteneşte şi deci este mai greu de testat şi întreţinut.

- **Cuplaje între clase de obiecte (Class Coupling);**

Această metrică va calcula de câte alte clase este legată o anumită clasă. Prea multe cuplaje împiedică modularitatea şi refolosirea codului.

Se măsoară, numărând clasele distincte, care nu sunt moştenite, de care depinde o altă clasă. O clasă este cuplată de alta, dacă apelează funcţiile sau variabilele acesteia.

- **Linii de cod (Lines of Code);**

Acesta metrică este simplă şi foarte cunoscută. Rezultatul obţinut cu ajutorul Visual Studio .NET trebuie utilizat cu atenţie, în mod ideal doar pentru informare, pentru a vedea cât de mare este o clasă sau o metodă.

Pragurile de alarmare pentru aceste metrici, conform convenţiei specialiştilor de la Microsoft sunt următoarele:

Tabel 2.2.1. Metrici interne şi limitele lor [Swm9-3]

Metrica	**Prag**
Adâncimea arborelui de moştenire al unei clase	Avertizare dacă este peste 5 nivele de adâncime
Complexitate ciclomatică	Avertizare dacă depăşeşte valoarea 25
Index de mentenabilitate	Avertizare dacă scade sub valoarea 20
Cuplaje între clase de obiecte	Avertizare dacă este peste 80 pentru o clasă şi peste 30 pentru o metodă

2.3. Metode şi tehnici de asigurare a calităţii sistemelor software

2.3.1. Prevenirea injectării defectelor în software

Prevenirea injectării defectelor prin blocarea sau eliminarea surselor de eroare are la bază următoarele:

- Eliminarea anumitor surse de eroare, cum ar fi: comunicarea ambiguă, neînţelegerea cerinţelor, etc.
- Prevenirea sau blocarea greşelilor prin corectarea sau blocarea directă a erorilor umane. În acest scop se pot folosi instrumente şi tehnologii, standarde de proces şi produs, etc.

Metodele prin care se poate preveni injectarea defectelor software sunt:

- **Educaţie şi instruire**

Această metodă are ca scop cunoaşterea tipului produsului şi a domeniului specific produsului, cunoaşterea şi expertiza în dezvoltarea de software (lipsa de expertiză în analiza cerinţelor şi specificarea software poate conduce la multe probleme în fazele următoare), cunoştinţe despre metodologia de dezvoltare, tehnologia şi instrumentele de dezvoltare, cunoaşterea procesului de dezvoltare (neînţelegerea procesului incremental poate conduce la multe probleme de interfaţare sau interacţiune).

- **Metode formale**

Aceste metode permit eliminarea unor surse de eroare şi verificarea absenţei greşelilor corelate. Metodele de dezvoltare formală presupun: specificarea formală (sunt specificate formal funcţiile produsului, constrângerile de mediu şi de proiectare, reducându-se şansa injectării de greşeli accidentale), verificarea formală (se verifică conformitatea proiectului software sau a codului faţă de specificaţia formală).

Limitarea utilizării metodelor formale este dată de următoarele aspecte: dificultatea specificării formale (cost asociat) şi necesitatea instrumentelor automate de suport.

- **Alte metode**

Alte metode presupun utilizarea de metodologii sau tehnologii software care reduc şansele de injectare a greşelilor. Un proces de dezvoltare bine gestionat presupune: planificarea procesului, gestiunea configuraţiilor şi a schimbărilor, utilizarea de instrumente software în procesul de dezvoltare, etc.

2.3.2. Tehnici de eliminare a defectelor

Principalele tehnici de eliminare a defectelor, înainte ca produsul software să fie folosit de către utilizatorul final sunt:

- Activităţile de inspecţie – se detectează şi se elimină greşelile din codul sursă, documentele de proiectare şi specificare;
- Testarea – se elimină greşelile pe baza căderilor constatate în timpul execuţiilor programului;

Inspecţii: detecţia şi eliminarea directă a defectelor

Inspecţiile software reprezintă examinări critice ale artefactelor software de către inspectori umani, cu scopul descoperirii şi reparării greşelilor din produsul software. Aceste inspecţii constă în citirea şi analiza codului şi artefactelor de specificare, proiectarea, planificarea testelor, etc. Greşelile sunt detectate fie direct de către inspectori, fie în cadrul unor întâlniri de grup. Greşelile identificate trebuie să fie eliminate ca rezultat al inspecţiei iar eliminarea lor este de asemenea verificată. Există diferite procese de inspecţie, dar în mod tipic ele includ planificare şi alte activităţi care urmează activităţilor de inspecţie propriu-zise.

Testare: observarea căderilor şi eliminarea greşelilor identificate

Testarea reprezintă cea mai importantă activitate de QA (quality assurance - asigurarea calităţii). Este efectuată la diferite nivele ale produsului (unitate, componentă, subsistem, sistem) şi în diferite faze ale dezvoltării. Testarea *Black-box* verifică efectuarea corectă a funcţiilor externe oferite de software în timp ce testarea *White-box* verifică implementarea corectă a unităţilor interne, a structurilor şi relaţiilor dintre ele. Testarea se opreşte în funcţie de diverse criterii de acoperire:

- Checklists: testarea funcţiilor majore şi a principalelor scenarii de utilizare;
- Acoperirea codului: execuţia tuturor instrucţiunilor, traversarea tuturor ramificaţiilor, ș.a.;
- Obţinerea fiabilităţii cerute (în condiţiile de utilizare);

În continuare se vor detalia metode pentru asigurarea calităţii prin tehnici de toleranţă la defecte.

2.3.3. Tehnici de toleranță la defecte

Datorită complexităţii şi dimensiunii multor sisteme software actuale, metodele de prevenire şi reducere a defectelor nu pot elimina toate defectele: numărul de teste necesare ar putea fi prea mare.

Pentru sisteme software ale căror căderi au un impact mare (sisteme de control în timp real utilizate în aplicaţii medicale, nucleare, transport, sisteme incorporate, etc.), defectele rămase pot fi diminuate prin tehnici de izolare.

Tehnicile de toleranţă la defecte au ca scop menţinerea sistemului în stare operaţională (posibil la o capacitate redusă) în cazul unei căderi locale.

Empiric s-a observat că echipamentele tind să aibă o mortalitate care urmăreşte o curbă, ilustrată în figura 2.3.3.1: sistemele foarte tinere şi cele foarte uzate se strică mult mai des decât sistemele „mature". „Burn-in" este o fază de testare care foloseşte componentele până acestea devin mature; în acest fel, componentele cu mortalitate infantilă ridicată sunt eliminate.

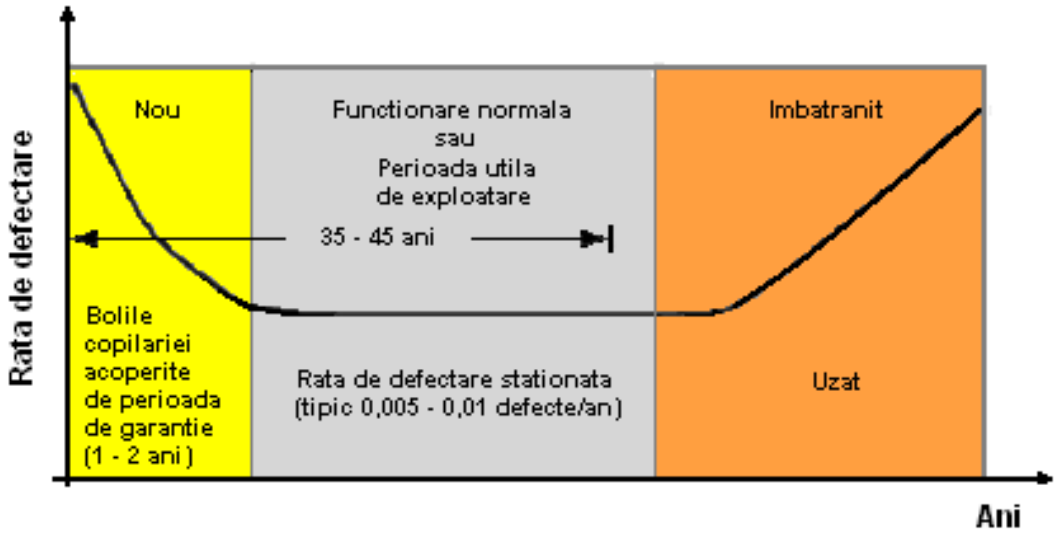

Fig. 2.3.3.1. Graficul ratei de defectare a unui echipament hardware/electric în funcţie de vârsta sa

Se observă că acest grafic are forma unei „căzi de baie" sau albie, de aceea mai este denumit şi „bathtub curve". Interpretarea graficului ne furnizează informaţia că echipamentele foarte noi şi foarte vechi au o probabilitate mai mare de a se defecta.

Fiabilitatea software, nu are aceleaşi caracteristici ca şi fiabilitatea hardware. Un exemplu de curbă ce modelează rata de defectare pentru un sistem software este prezentată în figura 2.3.3.2., împreună cu un exemplu de cum este influenţată această rată de către modificări în cadrul software-ului.

Există două diferenţe majore între curbele de fiabilitate hardware şi software. O diferenţă este că în ultima fază, software-ul nu are o rată de defectare din ce în ce mai

crescută, în ce timp ce hardware-ul are. În această fază, software-ul se apropie de uzura morală, nu există motivaţie pentru orice modificări ale software-ului. Prin urmare, rata de defectare nu se va schimba. A doua diferenţă este că software-ul va cunoaşte o creştere drastică ca şi rată de defectare de fiecare dată când se face o modificare. Rata de defectare se nivelează treptat, în parte din cauza defectelor constatate şi remediate după modificări.

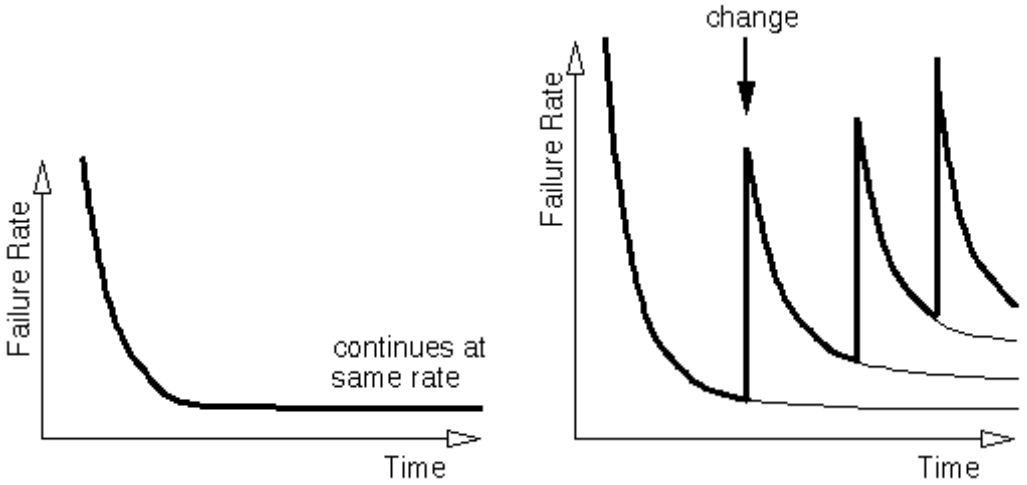

Fig. 2.3.3.2. Graficul ratei de defectare pentru un produs software

Schimbările din figura de mai sus implică modificări de funcţionalităţi, nu modificări pentru îmbunătăţirea fiabilităţii. Pentru upgrade-uri legate de îmbunătăţirea funcţionalităţii, complexitatea software-ului este posibil să crească. Chiar şi remedieri de bug-uri pot fi motiv pentru căderi ale software-ului, în cazul în care repararea bug-urilor introduce alte defecte în software.

Pentru upgrade-uri destinate creșterii fiabilităţii, cum ar fi reproiectarea sau reimplementarea unor module folosind tehnici mai bune de dezvoltare software, este posibil ca rata de defectare să scadă.

Tehnicile de toleranţă la defecte pornesc de la premiza existenţei defectelor în diferite componente ale unui sistem, scopul lor fiind de a menţine sistemul în funcţiune în cazul apariţiei unui defect.

Se iau în considerare două ipoteze:

1) Ipoteza unui eveniment rar

Unele căderi ale sistemelor şi accidentele sunt asociate cu evenimente rare, cu probabilităţi extrem de scăzute. Prin urmare, este imposibil să se anticipeze toate aceste evenimente rare, altfel sistemele noastre ar fi proiectate şi puse în aplicare pentru a le face

faţă. În consecinţă, sunt necesare acţiuni dinamice în timpul operării sistemelor pentru a trata problemele asociate cu astfel de evenimente rare. Aceste acţiuni dinamice constituie o mare parte din munca în toleranţă la defecte şi izolare a căderilor pentru sistemele software.

2) Ipoteza independenţei căderii

Subsistemele sau componentele se defectează independent una de cealaltă. Pornind de la această presupunere, sunt utilizate duplicarea sau alte tehnici de asigurare a siguranţei (Safety Assurance Techniques) care funcţionează eficient.

În cazul sistemelor software pentru care impactul căderilor este substanţial (De ex. cele care asigură infrastructura de telecomunicaţii la nivel mondial, financiar şi baze de date critice pentru companii mari, sisteme software de control în timp real utilizate în domeniul medical, nuclear, de transport şi sisteme embedded), riscul de cădere nu poate fi tolerat. O soluţie la astfel de probleme este duplicarea şi backup-ul pentru a reduce şansele unor căderi software sau daune din cauza lor.

Generarea unui număr mare de cazuri de testare pentru a acoperi toate aceste condiţii sau de a efectua operaţiuni de verificare bazată pe analiza tuturor scenariilor posibile este foarte dificil de realizat. În schimb, alte mijloace trebuie să fie utilizate pentru a preveni căderile prin ruperea relaţiilor de cauzalitate între aceste defecte şi căderile care rezultă. În acest fel, se pot „tolera”' aceste defecte, sau sunt constrânse căderile pentru a reduce prejudiciul rezultat. În mod similar, atunci când sunt implicate componente software duplicate, ar trebui să se asigure independenţa acestora (conform ultimei ipoteze – ceea a independenţei căderii), astfel încât fiabilitatea sau siguranţa sistemului global să poată fi îmbunătăţită.

Principalele tehnici de toleranţă la defecte sunt:

- Recovery blocks (blocuri de recuperare);
- N-version programming (programare cu N versiuni);
- Self-checking (auto-verificarea);

2.3.3.1. Utilizarea blocurilor de recuperare (recovery blocks)

Utilizând procesoare din ce în ce mai puternice şi mai rapide, putem repeta anumite sarcini de calcul într-un termen stabilit, fără a afecta grav performanţa sistemului. În acest caz, putem folosi blocuri de recuperare în mod repetat pentru a stabili puncte de control (checkpoints) şi a repeta paşii de calcul atunci când pot apărea sau sunt observate probleme în timpul operării.

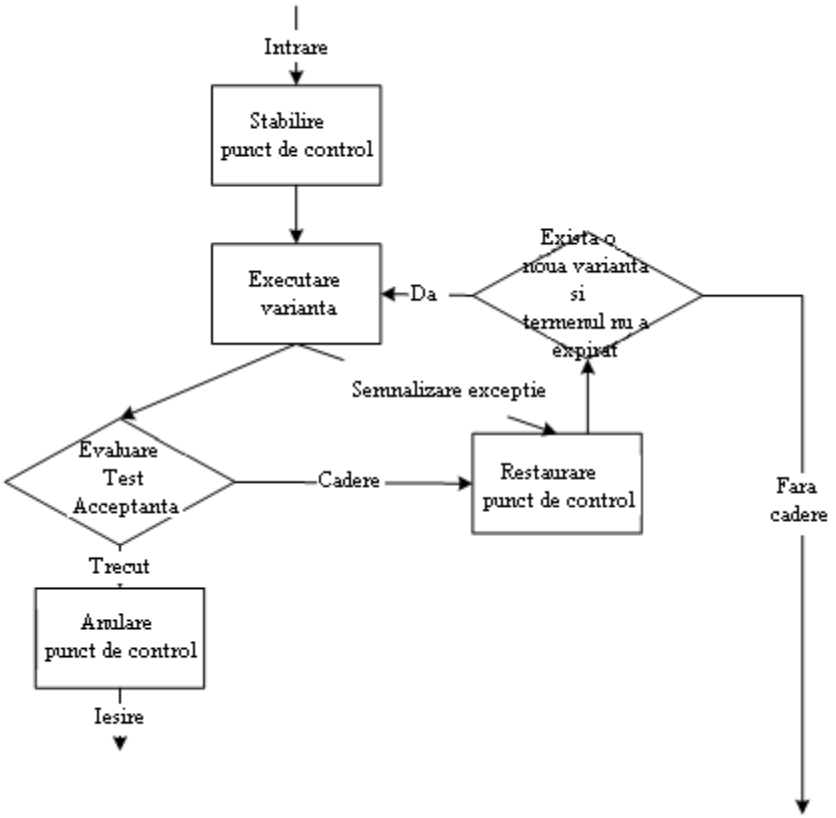

Fig. 2.3.3.1.1. Structura şi operarea blocurilor de recuperare

Conform [Swq9-9], un bloc de recuperare este executat prin efectuarea fiecărei variante pe rând, începând cu varianta principală, până când pentru o anumită variantă, testul de acceptanţă este satisfăcut. Executarea fără erori a unei variante este urmată de evaluarea făcută prin testul de acceptanţă. Dacă această evaluare este fără erori (ce impune ca testul să aibă valoarea adevărat), atunci testul de acceptanţă a fost îndeplinit şi execuţia blocului de recuperare este completă. În caz contrar, va fi semnalată o stare de eroare, la care sistemul răspunde prin restaurarea programului în starea de dinaintea intrării în varianta principală. Execuţia continuă apoi cu următoarea variantă, dacă aceasta există. Dacă, totuşi toate variantele au fost încercate şi niciuna nu a trecut testul de acceptanţă, va fi semnalată o stare de eroare către blocul de recuperare, prin care se indică că orice recuperare va fi efectuată doar prin ataşarea unui nou bloc de recuperare.

Utilizarea recovery blocks presupune introducerea duplicărilor în execuţia software-ului pentru ca defectele ce pot fi cauzate de rularea diverselor variante să producă doar o pierdere parţială a datelor obţinute prin execuţie şi nu căderi ale software-ului.

2.3.3.2. Folosirea NVP (N-version programming)

Domeniul ingineriei software include metode prin care se poate cuantifica şi îmbunătăţi calitatea programelor. Una dintre soluţiile studiate are legătură cu tehnicile de votare folosite pentru toleranţa erorilor hardware. Numele acestei soluţii este „programarea cu N versiuni” (NVP) [Swq9-7]. Tehnica votării foloseşte redundanţa: dispozitivul de calcul sau programul software este replicat de N ori şi rezultatul final obţinut este cel obţinut în majoritatea cazurilor individuale.

Bug-urile software sunt persistente: aflat în aceleaşi condiţii, programul se va comporta în acelaşi fel. Tehnicile de votare sunt neputincioase dacă toate componentele returnează aceeaşi eroare în acelaşi timp. Programarea cu N versiuni se realizează prin executarea în paralel a N programe diferite, scrise de echipe diferite de programatori, dacă e posibil, folosind instrumente şi tehnologii diferite. Toate cele N programe rezolvă aceeaşi problemă, dar în moduri diferite. Folosind o astfel de strategie, tehnica votării poate funcţiona în cazul programelor.

Specificaţii imprecise ale problemei pot fi detectate cu uşurinţă de această tehnică, pentru că implementările diferite pot lua decizii diferite pentru cazurile nespecificate clar. Din nefericire, programarea cu N versiuni este o metodologie foarte scumpă, folosită numai pentru aplicaţii critice, unde siguranţa este fundamentală.

Metoda NVP este, în general, mai adecvată faţă de metoda blocurilor de recuperare (recovery blocks) atunci când deciziile în timp util sau de performanţă sunt critice, cum ar fi în multe sisteme de control în timp real, sau când defectele software, în locul perturbaţiilor de mediu, sunt mai degrabă sursele primare de probleme.

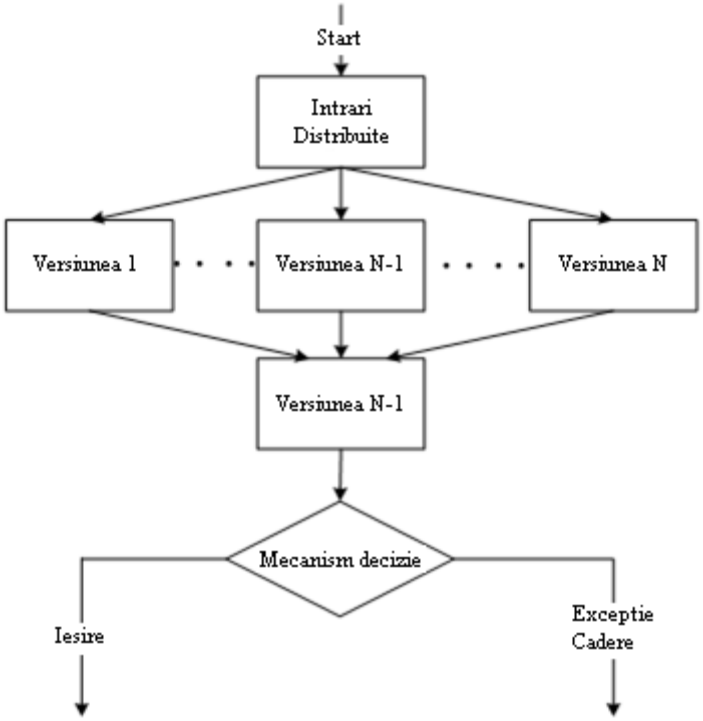

Fig. 2.3.3.2.1. Structura NVP

2.3.3.3. Self-checking

Metoda self-checking (software cu auto-verificare) nu este o metodă riguros descrisă în literatura de specialitate [Swq9-8], ci mai degrabă o metodă ad-hoc folosită în unele sisteme importante.

Software-ul cu auto-verificare include controale suplimentare, inclusiv mai multe puncte de verificare (checkpoint) şi metode de recuperare (rollback) introduse în sisteme tolerante la defecte sau sisteme critice. Alte metode includ taskuri separate, care acţionează în stivă (heap), găsind şi corectând defecte de date. În timp ce auto-verificarea nu poate fi o metodologie riguroasă, aceasta s-a dovedit a fi surprinzător de eficientă.

Problema evidentă la software-ul cu auto-verificare este lipsa de rigoare. Acoperirea codului pentru un sistem tolerant la defecte este necunoscută. Totuşi, cât de fiabil este un sistem realizat cu software-ul cu auto-verificare? Fără rigoarea corespunzătoare şi fără experimente, compararea şi îmbunătăţirea software-ul cu auto-verificare nu pot fi realizate în mod eficient.

În [Swq9-10], [Swq9-11], [Swq9-12], s-a propus o implementare a acestei metode.

Ideea de bază este că un program ar trebui să își verifice rezultatele (datele de ieşire) prin efectuarea de calcule redundante. Chiar dacă acestea folosesc acelaşi algoritm, în cazul

în care programul este „aproape corect”, este foarte puţin probabil ca după o secvenţă de verificări cu rezultate corecte, totuşi acestea să fie eronate.

Există totuşi un impediment serios pentru auto-verificarea la momentul execuţiei: dacă un program descoperă o inconsistenţă în verificări, nu se poate concluziona că programul este „aproape corect”. În loc de raportarea acestui rezultat, se poate concluziona că nu trebuie avută încredere în rezultatele obţinute.

Auto-verificarea reprezintă un punct de vedere destul de diferit faţă de testarea normală, deoarece descrie o abordare înţeleaptă din punct de vedere al calităţii. Încercările de testare pentru a prezice comportamentul viitor al unui program în mod uniform se fac pentru toate intrările posibile. În [Swq9-10], autorul se mulţumeşte să realizeze predicţia o dată la fiecare execuţie. Prin urmare, pentru a fi util, calculul trebuie să se facă la momentul execuţiei, atunci când scopul este cunoscut.

Testarea pentru a prezice uniform comportamentul are de suferit din cauză că la un program ce posedă o calitate software ridicată, erorile sunt foarte greu de găsit.

Eşantioanele mari de date sunt semnificative dar nu sunt practice, fiind greu de manevrat. Acest aspect poate fi folosit şi într-un mod avantajos, deoarece erorile sunt puţin probabile, calculele pot fi verificate cu acelaşi program [Swq9-12].

Rezultatele vor fi probabil aceleaşi, cu excepţia cazului în care sunt toate greşite şi asta datorită faptului că un rezultat greşit este aproape imposibil de replicat.

Tehnicile enumerate anterior sunt cele clasice în ceea ce priveşte utilizarea lor pentru toleranţă la defecte.

În ultima perioadă au fost dezvoltate şi alte tehnici cum ar fi:

- Reconfigurare şi reîntinerire;
- BASE;

2.3.3.4. Reconfigurare şi reîntinerire

Reconfigurarea şi reîntinerirea sunt variante complementare pentru software-ul tolerant la defecte. Reconfigurarea este reactivă în timp ce reîntinerirea este proactivă.

Reconfigurarea software-ului poate utiliza resurse redundante pentru recuperarea în timp real, în timp ce consideră, în mod dinamic, influenţa mai multor factori (serviciile sistemului de operare, încărcarea procesorului, memoria, etc.).

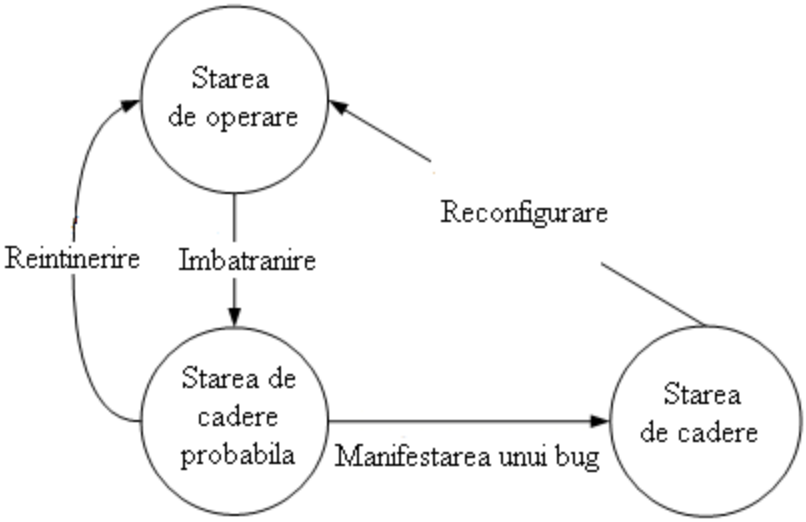

Fig. 2.3.3.4.1. Reconfigurarea şi reîntinerirea

Reîntinerirea este o abordare nouă pentru remedierea erorilor software datorate vechimii software-ului. Aceasta poate fi văzută ca o soluţie preventivă şi proactivă ce este utilă pentru împiedica fenomenul de îmbătrânire a software-ului.

Tehnica implică oprirea rulării software-ului la anumite momente de timp, „curăţarea" proceselor interne şi repornirea lui. Curăţarea proceselor interne ale unui software presupune refacerea spaţiului disponibil (garbage collection), curăţarea tabelelor kernel-ului sistemului de operare, reiniţializarea structurilor de date interne, etc.

Un bine cunoscut exemplu de reîntinerire care este utilizat des, îl reprezintă repornirea (resetarea) hardware a staţiei/calculatorului pe care rulează software-ul.

2.3.3.5. BASE

BASE (Byzantine Abstract Specification Encapsulation) este o tehnică de toleranţă la defecte, bazată pe BFT (Byzantine Fault Tolerance). BFT presupune dezvoltarea unui serviciu care să tolereze un comportament arbitrar faţă de replicarea defectelor, ca de exemplu comportamentul cauzat de un bug software sau un atac informatic.

Obiectivul BFT este de a proteja sistemul împotriva "căderilor bizantine", în care diverse componente cad sau se defectează în diverse moduri (obţinerea unor rezultate sau date de ieşire incorecte sau inconsistente, coruperea stării locale, procesarea incorectă a cererilor, etc.). Componentele ce funcţionează corect ale unui sistem bazat pe BFT pot reproduce serviciile sistemului presupunând că nu sunt prea multe componente cu defecte.

Spre exemplu, dacă în cadrul unei aplicaţii software, o funcţie depinde de rezultatele alteia şi acea funcţie furnizează rezultatele cu o mică abatere (eroare) atunci ea se va propaga prin intermediul celei de-a doua funcţii şi se va ajunge în final la un rezultat total eronat.

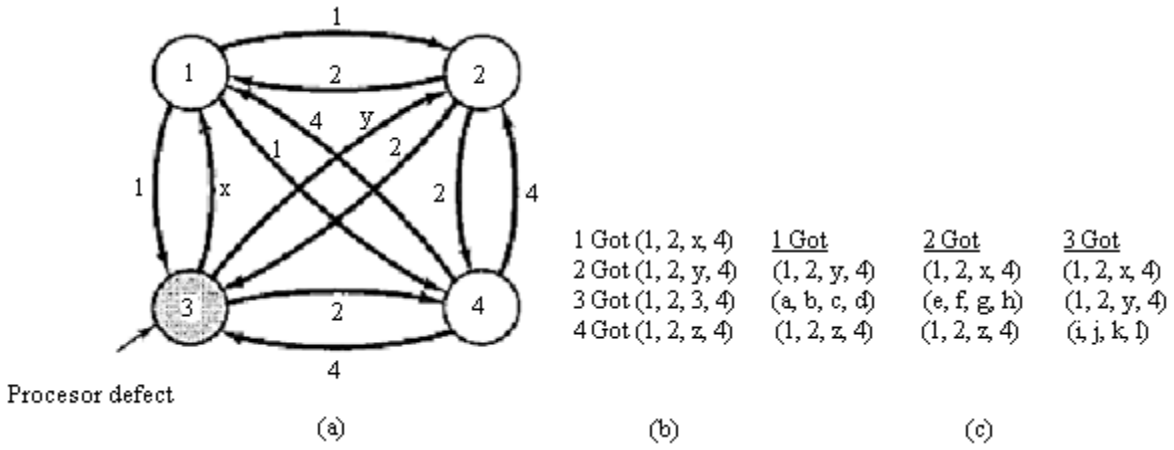

Fig. 2.3.3.5.1. Exemplificarea BFT prin algoritmul recursiv al lui Lamport (1982)

În cazul exemplificat în figura de mai sus, sunt utilizate 4 procesoare, din care unul singur transmite rezultate eronate (este defect). Acest lucru s-a verificat prin cereri de identificare emise de către fiecare din procesoare. La sfârşit s-au comparat răspunsurile primite şi astfel s-a identificat procesorul defect. Totuşi Lamport a demonstrat că pentru un sistem cu m procesoare defecte, acordul poate fi atins numai dacă 2m + 1 procesoare corect funcţionale sunt prezente, dintr-un total de 3m + 1. Acordul este posibil numai dacă mai mult de două treimi din procesoare lucrează corect [Swq9-13].

BFT presupune implementarea unui algoritm de replicare a stărilor unui sistem (state machine replication algorithm). În cazul în care un singur server centralizat implementează un serviciu, rezultatul va fi tolerant la defect atât cât va fi şi procesorul ce execută serviciul de pe server. Dacă acest nivel de toleranţă la defecte este inacceptabil, atunci se vor folosi servere multiple care se pot defecta independent. De obicei, replicări ale unui singur server sunt executate pe procesoare separate într-un sistem distribuit şi se folosesc protocoale pentru a coordona interacţiunile clientului cu aceste replicări.

BASE reduce costurile deoarece activează refolosirea implementărilor de servicii autonome. Îmbunătăţeşte disponibilitatea deoarece fiecare replică poate fi reparată periodic folosind un punct de vedere abstract al stării salvate de către replicările corecte şi deoarece fiecare replicare poate rula implementări de servicii distincte şi nondeterministe, ce reduc probabilitatea unor căderi obişnuite.

2.3.3.6. Securitatea şi integritatea transmisiei datelor

Deoarece sistemele de monitorizare şi control a staţiilor electrice, utilizează o arhitectură client-server, un alt aspect important de asigurare a calităţii este securitatea transmisiei datelor între server şi clienţi.

Pentru o creştere a securităţii transmisiei datelor se poate folosi protocolul IPv6. Internet Protocol version 6 IPv6) reprezintă generaţia următoare de protocoale folosite în Internet pentru reţelele cu comutare de pachete [Net9-1]. IPv4, versiunea folosită pe scară largă, tinde să nu scaleze, numărul de staţii conectate la o astfel de reţea fiind limitat. IPv6 implică în primul rând mărirea enormă a spaţiului de adrese dar şi autoconfigurarea adresei printr-un mecanism fără stări, standardizarea dimensiunii unei subreţele şi integrarea securităţii din protocolul IPsec.

Datorită dimensiunii mari a spaţiului unei adrese IPv6, scanarea aleatoare după sisteme ce sunt vulnerabile este complet inutilă. Atunci când ameninţările de tip malware erau de actualitate, acum câţiva ani, un sistem Windows fără patch-uri instalate ar fi fost infectat mai rapid, chiar înainte de apariţia update-urile de securitate necesare. Cu IPv6 acest lucru este pur şi simplu imposibil: chiar şi cu un miliard de host-uri infectate fiecare scanând un miliard de adrese IPv6 pe secundă, durează foarte mult numai scanarea spaţiului adreselor IPv6 alocate ISP-urilor (furnizorii de servicii Internet - Internet Service Provider). Scanările targeted, deşi nu sunt uşoare, sunt încă posibile astfel că măsuri de securitate precum cele folosite pentru IPv4 sunt încă necesare [Net9-2].

Este absolut necesară asigurarea integrităţii datelor transmise între echipamentele hardware de monitorizare şi aplicaţiile software ce achiziţionează informaţii de la acestea (aplicaţiile tip server). Una dintre metodele propuse este utilizarea CRC - Cyclic Redundancy Check (Control Redundant Ciclic). CRC este o formă de sumă de control, ce se bazează pe teoria polinoamelor de lungime maximă. Chiar dacă metoda CRC este mai sigură decât metoda bazată pe o simplă sumă de control, nu oferă o adevărată securitate criptografică. CRC este o tehnică folosită pentru detecţia erorilor de transmisie. Pentru detecţia şi corectarea erorilor, există un registru special în care se stochează suma de control a datelor transferate. Aceasta se compară cu suma de control calculată şi se elimină astfel posibilele erori. În acest caz, tehnica CRC este folosită doar pentru a asigura integritatea datelor la transferurile pe magistrală, nu şi pentru a îmbunătăţi integritatea datelor stocate pe hard-disk-uri.

2.4. Concluzii

În cadrul acestui capitol sunt descrise aspecte generale privind calitatea software şi metodele şi tehnicile utilizate pentru asigurarea acesteia.

Este descris pe scurt standardul ISO9126 şi modelul de calitate software împărţit în 6 caracteristici de calitate şi 21 de subcaracteristici.

Sunt descrise categoriile în care se împart metricile software: de dimensiune, complexitate, de calitate a produsului şi respectiv metrici ale procesului. Sunt enumerate metricile software incluse în pachetul Visual Studio .NET 2010.

Sunt prezentate tehnicile de asigurare a calităţii software care presupun utilizarea de tehnici şi metode de prevenire a injectării defectelor, de eliminare a defectelor şi de izolare a acestora.

Sunt descrise activităţile de inspecţie şi testare, utilizarea blocurilor de recuperare, N-version programming, self-checking precum şi tehnici mai noi ca reconfigurare şi reîntinerire, BASE.

Au fost prezentate pe scurt aspecte de asigurare a siguranţei transmisiei datelor în cadrul unei staţii electrice, ce implică utilizarea protocolului IPv6 pentru calculatoarele server şi respectiv client din cadrul reţelei ce există într-o staţie electrică. Pentru asigurarea integrităţii transmisiei datelor între IED-uri, se utilizează metoda CRC.

3. Analiza aspectelor de calitate pentru un sistem de monitorizare şi control a unei staţii electrice

3.1. Arhitectura generală pentru un sistem de monitorizare şi control a unei staţii electrice

Un sistem de monitorizare şi control a unei staţii electrice presupune existenţa unor echipamente electronice care monitorizează parametrii echipamentelor (denumite IED: Intelligent Electronic Device), unul sau mai multe calculatoare (servere) care centralizează informaţiile monitorizate şi le salvează într-o baza de date şi unul sau mai multe calculatoare pe care se executa o aplicaţie client cu rol de prezentare a parametrilor monitorizaţi.

Un astfel de sistem are, în general, o arhitectură client-server. Aplicaţiile server achiziţionează de la IED-uri, în timp real valorile parametrilor monitorizaţi ai echipamentelor electrice, le prelucrează şi apoi le salvează într-o bază de date locală sau centrală. Aplicaţia client achiziţionează informaţii din baza de date şi le prezintă utilizatorului (personalul din staţia electrică) printr-o interfaţă grafică prietenoasă (GUI – Graphical User Interface).

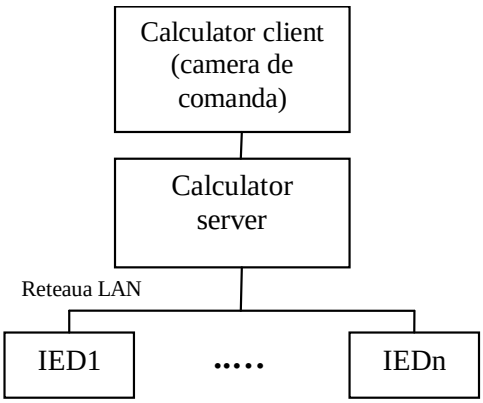

Fig. 3.1.1. Arhitectura generală a unui sistem de monitorizare şi control a unei staţii electrice

Un astfel de sistem trebuie să îndeplinească cel puţin două dintre cele mai importante caracteristici de calitate ale unui sistem software: fiabilitatea şi uşurinţa de întreţinere (mentenabilitatea).

Obţinerea acestor caracteristici trebuie să fie urmărită pe parcursul întregului proces de dezvoltare al sistemului. În cadrul tezei, am propus soluţii pentru: prevenirea injectării defectelor software, aplicarea de metode şi tehnici de eliminare a defectelor din produsul software şi asigurarea siguranţei în funcţionare prin tehnici de toleranţă la defecte.

3.2. Sisteme existente de monitorizare şi control a staţiilor electrice

În prezent, în domeniul monitorizării şi controlului staţiilor electrice, există câteva mari companii internaţionale de renume ce se ocupă de dezvoltarea hardware şi software a unor astfel de sisteme.

Sistemele de monitorizare pe care le-am considerat ca fiind reprezentative sunt cele dezvoltate de General Electric, ABB, Siemens şi AREVA.

Precizez că niciuna dintre aceste companii nu a implementat încă un sistem de monitorizare şi control complet a unei staţii electrice din România. Un astfel de sistem necesită investiţii financiare foarte mari atât din partea beneficiarului cât şi din partea producătorului.

Producătorii de echipamente electrice sunt avantajaţi în cazul în care dezvoltă un astfel de sistem deoarece pot integra şi controla mai uşor propriile echipamente.

Am avut ocazia să lucrez la un proiect pentru un astfel de sistem de monitorizare şi control complex, într-o staţie electrica aparţinând reţelei de transport a energiei electrice din România, dezvoltat de compania Nova Industrial.

3.2.1. Sistemul General Electric

Compania General Electric a dezvoltat un sistem de monitorizare şi diagnosticare denumit GE iSM&D (Integrated Substation Monitoring and Diagnostic). Prima versiune a acestui sistem a fost dezvoltată în anul 2001, iar ultima versiune, 3.0, în 2005.

Arhitectura sistemului iSM&D este prezentată în următoarea figură:

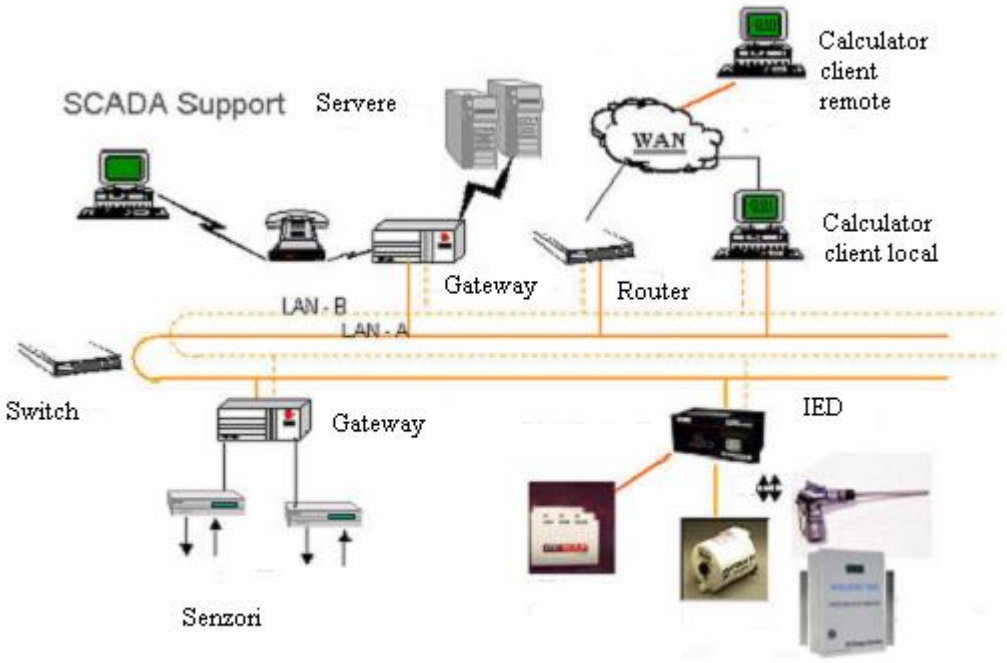

Fig. 3.2.1.1. Arhitectura sistemului iSM&D

Sistemul de monitorizare iSM&D se ocupă doar de monitorizarea transformatoarelor de putere dar poate fi prevăzut cu un modul pentru monitorizarea întreruptoarelor.

Arhitectura software este de tip client-server, clienţii din WAN accesând datele monitorizate prin intermediul unui browser web.

Aplicaţia software pentru monitorizarea transformatoarelor de putere (figura 3.2.1.2.) prezintă în partea superioară un meniu pentru diverse acţiuni ale utilizatorului (tipărire, raportare, rulare alte module, etc.), secţiunea detalii echipament, alarmare pentru componentele transformatorului, starea lui curentă, partea grafică reprezentând elementele componente transformatorului şi valori măsurate pentru parametrii corespunzători la momentul respectiv. Este disponibilă şi o secţiune de evenimente (alarmări) apărute într-un anumit interval de timp.

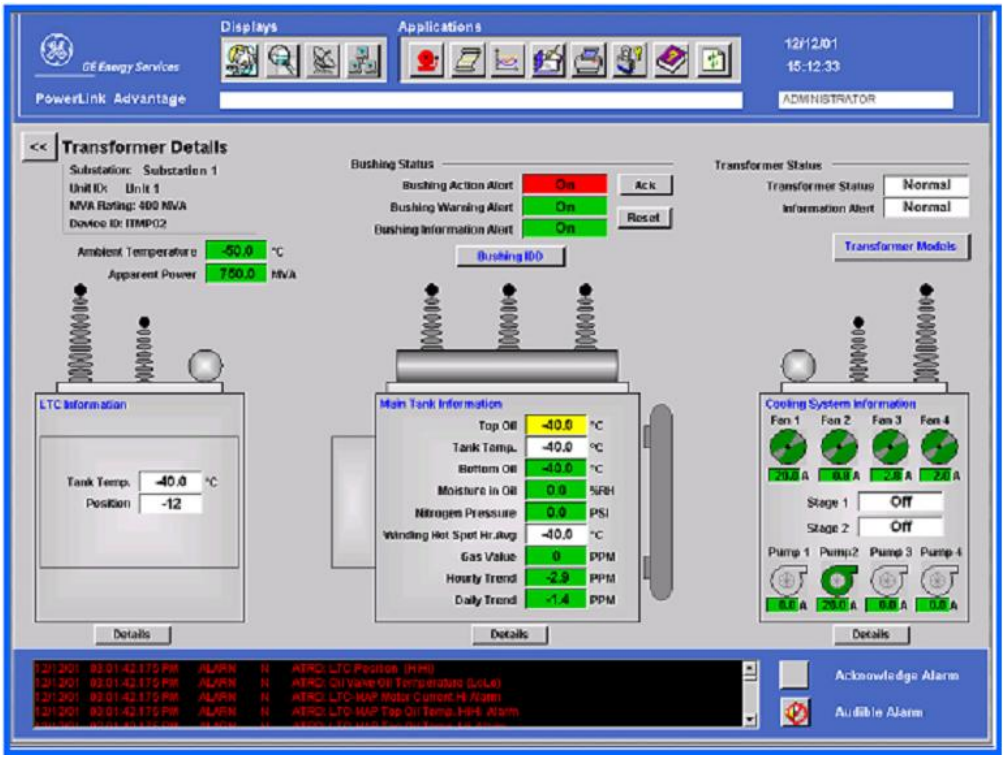

Fig. 3.2.1.2. Fereastra principală a aplicaţiei software din sistemul iSM&D

3.2.2. Sistemul ABB

Compania ABB a dezvoltat începând cu anul 2006 sistemul informatic SMS510 (Substation Monitoring System). Acesta efectuează monitorizarea principalelor componente pentru transformatoarele de putere din staţiile electrice precum şi a întreruptoarelor.

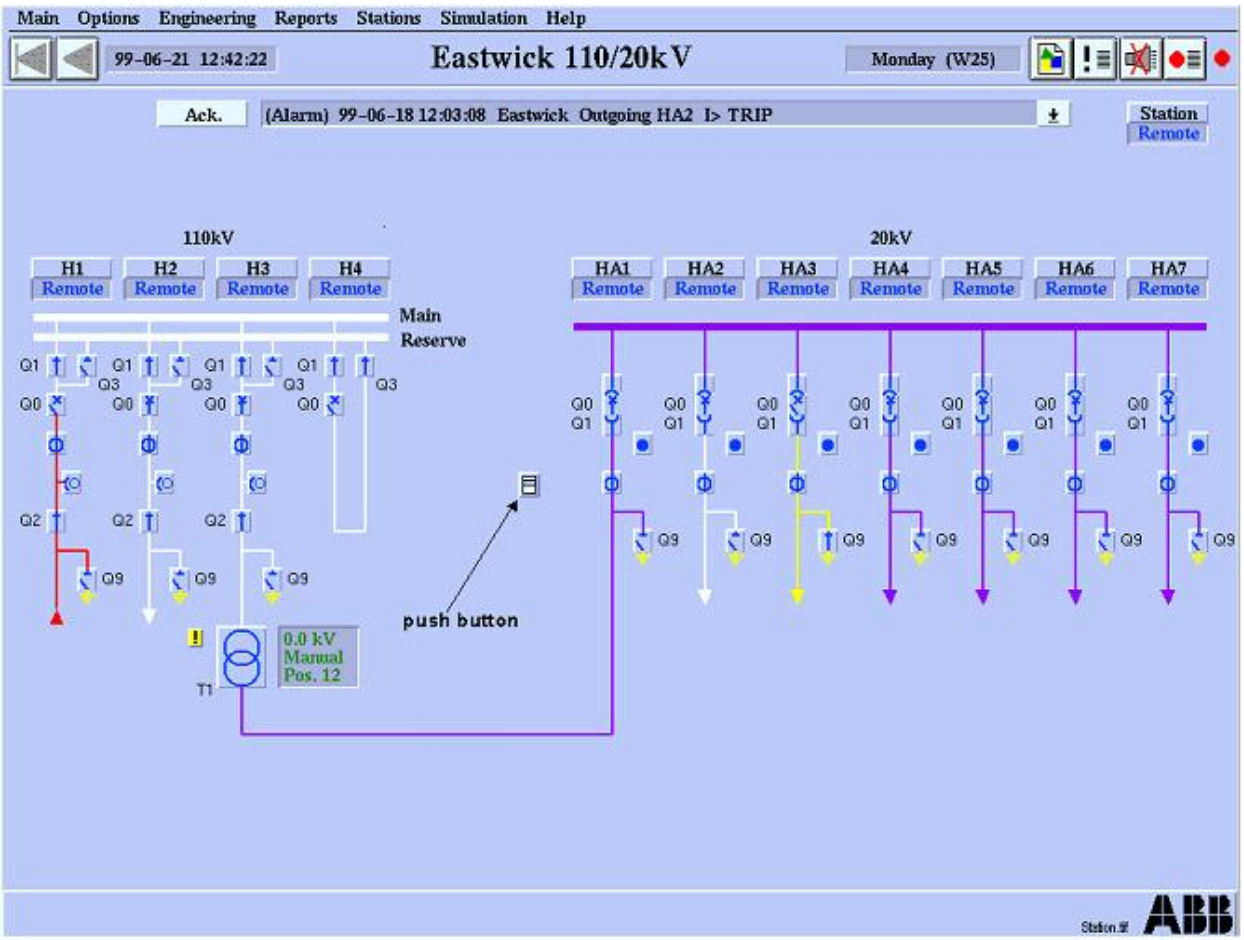

Fig. 3.2.2.1. Fereastra principală a sistemului de monitorizare SMS510

SMS510 oferă informaţii esenţiale despre procesele electrice de transport şi de distribuţie. Aceste informaţii cuprind datele măsurate, înregistrate şi calculate pentru echipamentele electrice monitorizate cum ar fi indicaţii, setări şi informaţii privind diagnosticarea, disponibile cu ajutorul IED-urilor.

SMS510 este un produs pentru monitorizarea staţiilor electrice care permite utilizatorilor să configureze sistemul pentru a-l utiliza cât mai facil.

3.2.3. Sistemul Siemens

Compania Siemens a dezvoltat sistemul informatic iSCM (Integrated Substation Condition Monitoring) care are în componenţă mai multe subsisteme pentru diferitele echipamente primare dintr-o staţie electrică:

- Monitorizarea transformatorului de putere;
- Monitorizarea întreruptorului;
- Monitorizarea separatorului;

- Monitorizarea descărcătorului;
- Monitorizarea transformatoarelor de măsura (curent şi/sau tensiune);
- Monitorizarea liniilor;
- Monitorizarea cablurilor;

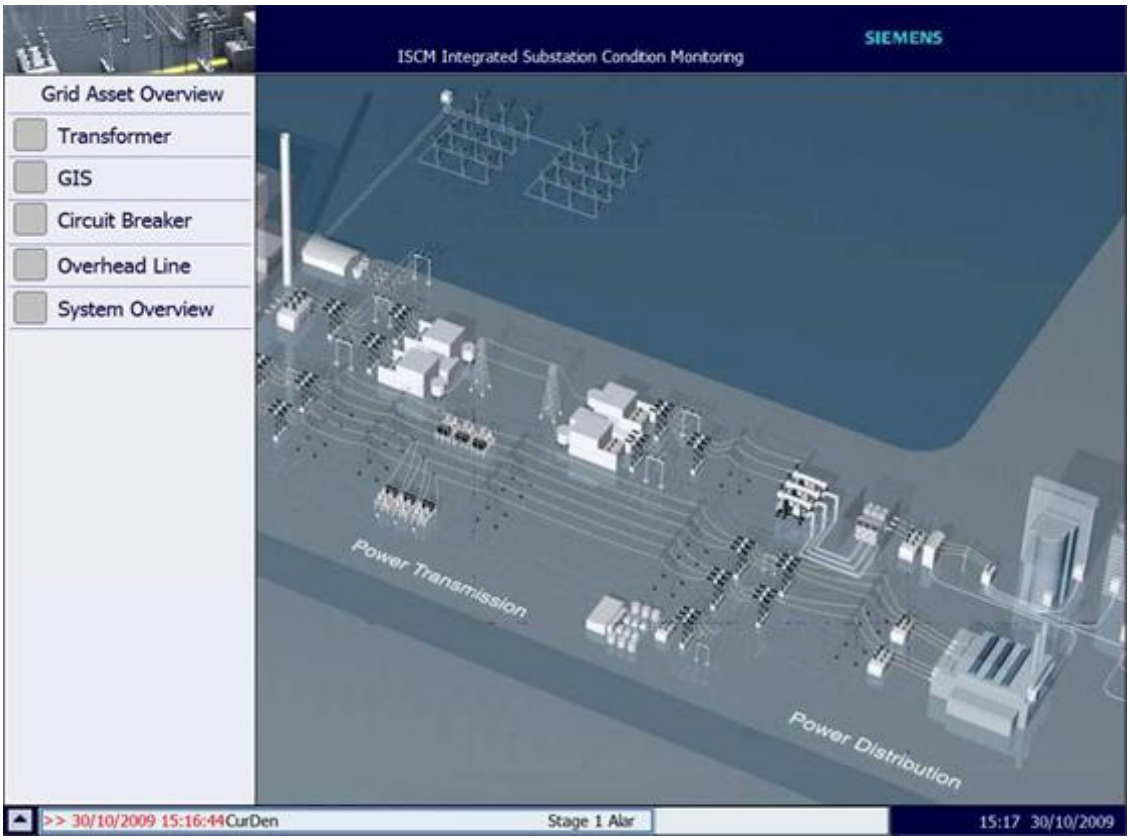

Fig.3.2.3.1. Sistemul de monitorizare Siemens iSCM

Din punct de vedere al echipamentelor primare monitorizate, sistemul de monitorizare Siemens iSCM este complet. Totuşi, nu există o aplicaţie software client care să preia informaţiile de la toate subsistemele de monitorizare a echipamentelor electrice.

Spre deosebire de alte sisteme complexe de monitorizare, nu există implementate toate subsistemele dedicate pentru fiecare tip de echipament electric, în cadrul aceleiaşi staţii electrice.

Datele monitorizate sunt colectate, analizate şi prezentate într-un format standard prin intermediul sistemului SCADA (supervisory control and data acquisition). Interfaţa cu utilizatorul prezintă informaţii clare, bine structurate, vizualizarea valorilor, fie numerice fie sub forma de grafice, curbe sau diagrame, înregistrarea rapoartelor, funcţii de căutare prin rapoarte.

3.2.4. Sistemul AREVA

PACiS (Protection, Automation&Control Integrated Solution) este sistemul de monitorizare complex on-line a staţiilor electrice dezvoltat de compania AREVA.

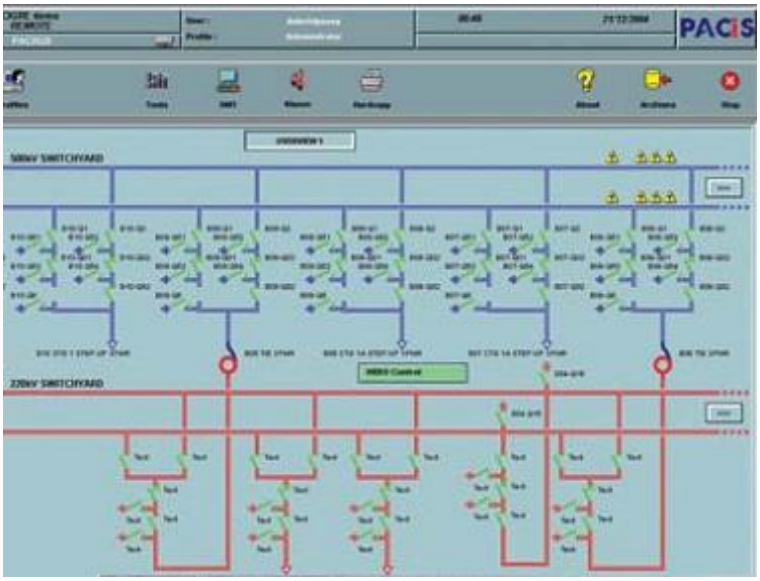

Fig.3.2.4.1. Sistemul de monitorizare AREVA PACiS

Arhitectura PACiS este structurată pe nivele ierarhice. Fiecare nivel menţine o performanţă dată în termeni de timp de transmisie, automatizare şi fiabilitate, independent de celelalte nivele. De aceea, se asigură că sistemul poate fi extins consecvent şi în siguranţă.

Arhitectura de bază PACiS interconectează un Remote Terminal Unit (RTU) sau un calculator, cu o serie de dispozitive electronice inteligente (IED). Aceasta arhitectură de tip client-server va fi folosită de obicei într-o staţie de distribuţie simplă, un parc eolian sau o staţie de transport.

Sistemul funcţionează în cadrul unei reţele de tip Ethernet şi include o aplicaţie software tip client ce prezintă interfaţa operatorului/utilizatorului (UI – user interface) şi IED-uri. Reţeaua Ethernet poate fi locală în cadrul unei staţii, de obicei pentru o aplicaţie de transmisie sau poate interconecta site-uri dispersate, cum există de obicei în aplicaţii industriale sau de infrastructură. Viteza de transfer mare prin Ethernet rezolvă problema blocajele de trafic de date întâlnite la protocoalele proprietare. Există posibilitatea de accesare de la distanţă a IED-urilor pentru operaţiunile legate de mentenanţă.

Modulele cheie ale interfeţei grafice sunt următoarele:

- Monitorizarea şi afişarea informaţiilor în timp real, ce include:
  - Diagrame pentru fiecare linie electrică;
  - Starea componentelor sistemului;
  - Alarme;
  - Secvenţe de evenimente;
- Partea de control cu ferestre specifice, care prezintă:
  - Secvenţe de selectare a componentelor înainte de punerea în funcţiune a sistemului;
  - Reprezentarea condiţiilor de acţionare a componentelor;
  - Permisiune de sincronizare-verificare;

3.2.5. Sistemul Nova Industrial

Nova Industrial este o companie românească ce a dezvoltat sistemul EMCSIT (Echipament pentru Monitorizarea Complexă a Staţiilor de Înalta Tensiune) pentru monitorizarea şi controlul staţiilor de înalta tensiune [Smg9-1].

EMCSIT este un sistem complex de monitorizare on-line a unei staţii electrice. Din punct de vedere hardware, sistemul este alcătuit din mai multe IED-uri (Intelligent Electronic Device) poziţionate în fiecare cabină de relee din cadrul staţiei, care sunt conectate la senzori şi traductori montaţi pe echipamentele electrice din staţie şi transmit informaţiile achiziţionate de la acestea către serverele locale.

Sistemul de monitorizare include aplicaţii de tip server ce se conectează la echipamentele de monitorizare (IED-uri); ele achiziţionează valorile parametrilor monitorizaţi şi le salvează într-o bază de date centrală, instalată pe un calculator server central. Acesta este accesat de aplicaţiile client EMCSIT din reţeaua locală.

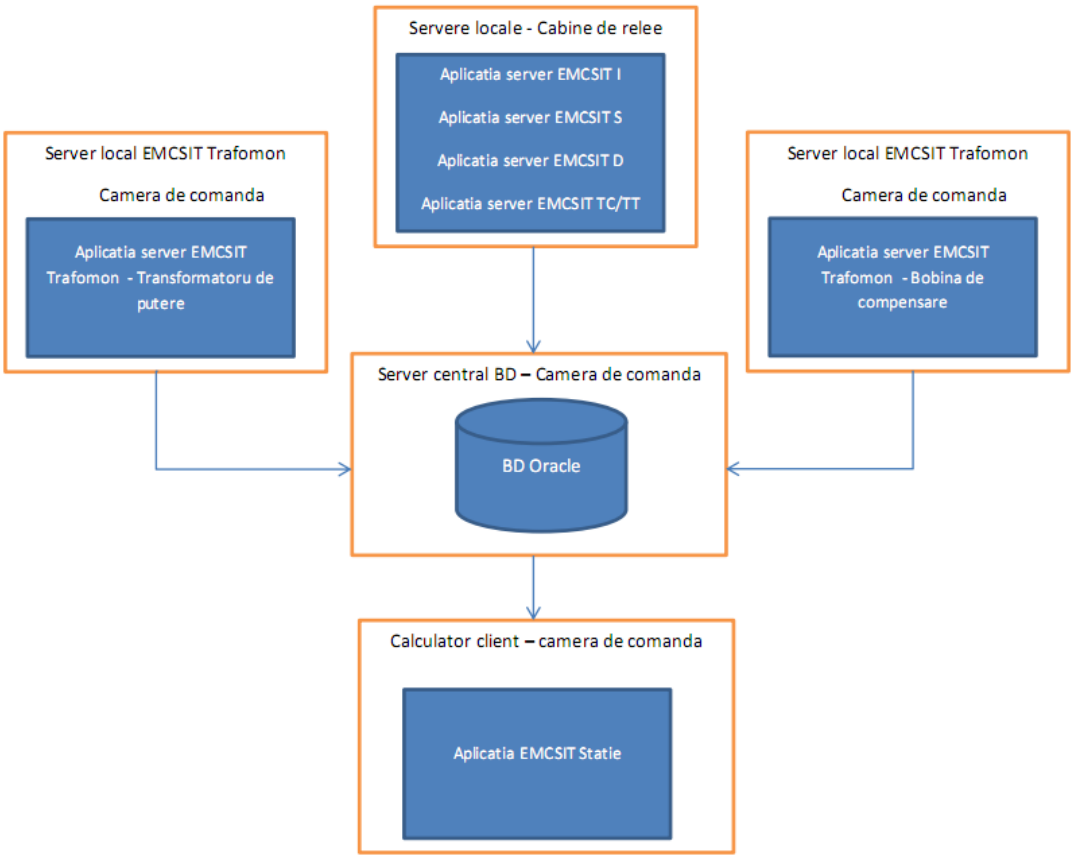

Fig.3.2.5.1. Arhitectura sistemului EMCSIT

Mărimile monitorizate sunt afişate în camera de comandă a staţiei electrice pe un calculator client. Timpul necesar pentru achiziţia informaţiilor de la aparatele de monitorizare şi afişarea valorilor în cadrul aplicaţiei trebuie să fie sub 100 de milisecunde.

Sistemul de monitorizare şi control EMCSIT include următoarele tipuri de aplicaţii: EMCSIT Server, EMCSIT Client şi EMCSIT Staţie:

Aplicaţiile “EMCSIT Server” care sunt instalate pe serverele locale:

- Sunt specifice fiecărui tip de echipament electric primar dintr-o staţie electrica (întreruptor, separator, descărcător, transformator de măsură de curent sau tensiune);
- Asigură achiziţia datelor de la echipamentul de monitorizare EMCSIT (IED), prelucrarea acestor date conform cu specificaţiile specialiştilor tehnologi din domeniu şi salvarea acestor date în baza de date locală;
- Aplicaţiile server sunt dezvoltate pentru întreruptoare (EMCSIT I Server), separatoare (EMCSIT S Server), transformatoare de

măsură de curent şi/sau tensiune (EMCSIT TC/TT Server) şi descărcătoare (EMCSIT D Server);

Aplicaţiile "EMCSIT Client" care sunt instalate pe calculatorul client din camera de comandă (sau pe orice calculator client) sunt dezvoltate pentru fiecare tip de echipament electric, şi au următoarele componente:

- Componenta de vizualizare a ultimelor date achiziţionate şi înregistrate de server;
  - EMCSIT - Client asigură preluarea datelor salvate în baza de date de către EMCSIT - Server, afişarea acestora pe ecranul principal al calculatorului client precum şi prezentarea alarmelor şi avertizărilor în caz de funcţionare eronată a aplicaţiei sau echipamentului electric monitorizat.

Sunt implementate următoarele verificări în software:

- Dacă există conexiune în reţea între calculatorul server central şi serverele locale;
- Dacă există conexiune la serverul central de baze de date;
- Diverse tipuri de avertizări în ceea ce priveşte erorile aparatelor de monitorizare (dacă este alimentat, dacă diversele componente interne funcţionează corect – memoria EEPROM, memoria flash internă, memoria flash externă, ceasul de timp real RTC, convertorul intern, etc.);

- Componenta de vizualizare şi analiză a evenimentelor înregistrate de echipamentul de monitorizare IED (pentru întreruptoare, separatoare şi transformatoare de măsură);
  - În cazul în care pentru echipamentul electric monitorizat se înregistrează evenimente în baza de date (ex. pentru întreruptor: închidere respectiv deschidere), EMCSIT - Grafice Evenimente, permite conectarea la baza de date, achiziţia informaţiilor înregistrate şi efectuarea calculelor parametrilor ce influenţează sau sunt influenţaţi de acel tip de eveniment.

Un eveniment pentru un echipament electric (în cadrul sistemului de monitorizare şi control EMCSIT, astfel de evenimente sunt înregistrate

pentru întreruptoare, separatoare, transformatoare de măsură de curent/tensiune) reprezintă o variaţie bruscă a parametrilor monitorizaţi. Aceasta variaţie este corespunzătoare unui eveniment fizic (acţionarea întreruptorului, deschidere/închidere separator, supracurenţi, supratensiuni) şi trebuie achiziţionată şi afişată rapid. Aceasta presupune o întrerupere a ciclului normal de achiziţie a informaţiilor de la echipamentele de monitorizare şi dedicarea unui fir de execuţie (thread) special pentru această operaţie.

- Componenta de vizualizare a istoricului datelor măsurate şi înregistrate de către serverele locale: EMCSIT Istoric. Acest modul are acces la datele înregistrate în baza de date de către serverele locale şi afişează istoricul pentru parametrii monitorizaţi ai echipamentelor electrice sub formă grafică sau tabelară.

Aplicaţiile client sunt dezvoltate pentru întreruptoare (EMCSIT I Client, Grafice evenimente, Istoric), separatoare (EMCSIT S Client, Grafice Evenimente, Istoric), transformatoare de măsură de curent şi/sau tensiune (EMCSIT TC/TT Client, Grafice Evenimente, Istoric) şi descărcătoare (EMCSIT D Client şi Istoric);

Aplicaţia "EMCSIT Staţie" prezintă schema monofilară completă pentru staţia electrică, afişând simboluri animate pentru echipamentele electrice primare monitorizate (dacă sunt în funcţiune, au tensiune: culoarea roşie; dacă nu funcţionează: culoarea verde) şi afişează valorile parametrilor monitorizaţi privind fiecare astfel de echipament. Schema este desenată dinamic, orice modificare a stării unui echipament electric monitorizat, fiind ilustrată corespunzător conform convenţiei cu beneficiarul în ceea ce priveşte simbolurile utilizate şi codul de culori. "EMCSIT Staţie" rulează pe calculatorul client din camera de comandă.

3.3. Smart Grid

Tendinţa internaţională, şi recent naţională, este de a crea reţele inteligente (Smart Grids) ce includ sisteme de monitorizare şi control în domeniul energetic. Investiţiile sunt masive şi realizabile pe termen mediu şi lung.

Deoarece nu există o organizaţie unică coordonatoare la nivel mondial, nu există o definiţie unică însuşită de specialiştii din întreaga lume. Majoritatea lumii ştiinţifice

mondiale acceptă că ”Reţelele/Reţeaua Inteligentă” vine de la termenul “Smart Grids/Grid”. Acesta a început să fie utilizat frecvent din 2003 în SUA şi din 2005 în Europa.

Indiferent de definiţia utilizată, o Reţea Inteligentă include un sistem automat de monitorizare şi control în timp real a lanţului producţie - consumator final de energie, folosind o reţea informatică şi de comunicaţii bidirecţionale. Smart Grid se bazează pe tehnologii de ultimă generaţie: supraconductivitatea, integrarea SER (surse de energie regenerabilă), utilizarea sistemelor expert în automatizarea, diagnosticarea şi conducerea instalaţiilor energetice, folosirea dispozitivelor electronice inteligente pentru aplatizarea curbei de sarcină şi alegerea tarifului.

În prezent, Reţeaua Inteligentă este definită ca un ansamblu de sisteme de control şi management al reţelei electrice, senzori şi mijloace de comunicare şi informare, care încorporează atât elemente tradiţionale cât şi de ultimă generaţie. Ea combină elemente de software şi hardware menite să îmbunătăţească semnificativ modul în care este condus/operat sistemul electric actual (de la cel de joasă tensiune până la cel de înaltă tensiune) şi să permită comunicarea în timp real între entităţile interesate din lanţul producţie - consumator final.

În cadrul Uniunii Europene (EU) se promovează, prin realizarea Planului Tehnologic Strategic (PTS), o politică energetică care să conducă la creşterea eficienţei energetice, accelerarea producţiei de energie regenerabilă, dezvoltarea tehnologiilor de tip Smart Grid în vederea obţinerii securităţii energetice, competitivităţii şi dezvoltării durabile. Având în vedere schimbările climatice, Comisia Europeana (CE) a propus:

- reducerea emisiilor de gaze cu efect de seră cu 20% pană în anul 2020 comparativ cu 1990;
- creşterea ponderii energiei regenerabile de la cca 11% în 2010 la 20% în 2020;
- reducerea consumului global de energie primară cu 20% până în 2020 prin introducerea de noi tehnologii şi creşterea eficienţei energetice;

În vederea realizării acestei directive un rol foarte important îl va avea implementarea Reţelelor Inteligente.

În ţara noastră, nu există un document elaborat la nivel de Guvern care să trateze strategia privind Reţelele Inteligente. Totuşi, România ca ţară membră UE trebuie să îndeplinească obiectivele stabilite de UE prin Directiva 28/2009 care prevede ca până în anul 2020, SER să aibă o pondere de 24% din totalul de consum al României.

Printre principalele măsuri pentru atingerea obiectivului de 24% este şi creşterea capacităţii reţelelor de distribuţie de medie şi joasă tensiune prin aplicarea tehnologiilor Smart Grid.

Conform „Strategiei Energetice a României pentru perioada 2007-2020” cele mai importante obiective care trebuie realizate sunt:

- Securitatea aprovizionării cu energie;
- Dezvoltarea durabilă;
- Competitivitatea.

Valoarea totală estimată a investiţiilor necesare realizării obiectivelor din Strategia 2007-2020 este de cca. 35 miliarde de euro.

În continuare se va prezenta modelul conceptual/cadru elaborat de Institutul Naţional de Standardizare şi Tehnologii - NIST [Smg9-9] pentru promovarea Smart Grid (Fig. 3.3.1.)

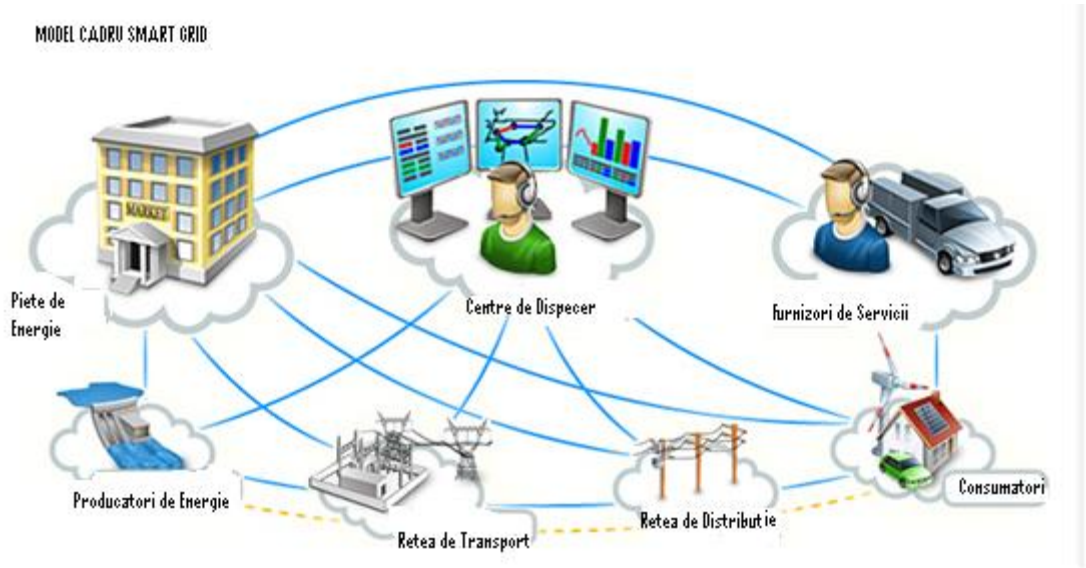

Fig.3.3.1. Model Conceptual SMART GRID

Reţeaua Inteligentă, prin caracterul interoperabil şi interactiv va permite companiilor de electricitate să livreze energie electrică în sistem cât mai economic şi eficient şi va permite tuturor categoriilor de consumatori să achiziţioneze energie electrică la preţuri mai reduse faţă de cele din 2010 pe termen mediu şi lung.

Realizarea acestui tip de reţea va contribui la reducerea emisiilor de gaze şi la o dezvoltare durabilă în condiţiile în care rezervele de energie primară din combustibili fosili sunt în scădere.

Analizând modelul american şi european privind realizarea Reţelelor Inteligente, se constată că una dintre verigile importante privind implementarea Smart Grid-urilor, ca parte componentă a infrastructurii instalaţiilor electrice, este staţia electrică. Pentru a contribui la realizarea acestui concept în România, a fost realizat proiectul EMCSIT - sistem complex de monitorizare on-line pentru o staţie electrică, indiferent de nivelul tensiunii şi complexitatea acesteia. În cadrul acestui proiect pilot sunt monitorizate toate echipamentele primare: transformatoare de putere, întreruptoare, separatoare, transformatoare de măsură (curent şi/sau tensiune) şi descărcătoare.

Pentru o tranziţie de succes a implementării tehnologiilor specifice Reţelelor Inteligente în prezent şi în viitor, toate părţile interesate trebuie să se implice efectiv: guverne, autorităţi de reglementare în domeniu, universităţi, entităţile de pe lanţul producţie - consumator final, producătorii de echipamente de instalaţii electrice, producătorii de dispozitive electronice inteligente (IED-uri), furnizorii de tehnologii informatice şi de telecomunicaţii. Coordonarea la nivel local, regional, naţional şi european este esenţială pentru realizarea obiectivelor prevăzute în Platforma Tehnologică Europeană privind realizarea Reţelelor Inteligente.

Implementarea cu succes a conceptului Smart Grid, ce include sisteme de monitorizare şi control la nivel de staţie electrică, va asigura nu numai energie electrică „curată" şi mai ieftină ci şi o dezvoltare durabilă.

3.4. Probleme care pot cauza căderi ale sistemului

Sistemul de monitorizare şi control poate cădea atunci când una dintre componentele sale, hardware sau software cade sau se defectează.

Componentele hardware incluse în IED-uri pot cădea din foarte multe motive, printre care: erori de proiectare a circuitelor electronice, utilizarea unor componente electronice care nu sunt fiabile, erori în realizarea circuitelor şi lipirea componentelor electronice, etc.

Componentele software sunt reprezentate prin: software-ul embedded din IED-uri şi aplicaţiile tip server respectiv client din componenţa sistemului de monitorizare şi control.

Software-ul embedded din IED-uri poate cădea din următoarele cauze:

- neacoperirea tuturor cazurilor ce impun restartarea sistemului şi blocarea într-o rutină ciclică care însă informează secţiunea de program ce se ocupă cu restartarea în cazul blocărilor (denumită şi watchdog) că programul rulează normal.

- coruperea memoriei EEPROM care ţine setările şi adresele fiecărui IED la o pornire cu tensiune scăzută sau oscilantă care nu acoperă minimul necesar pentru ca microcontroller-ul să efectueze operaţiile corect.

Aplicaţiile software tip server respectiv client pot avea următoarele cauze ale căderilor:

- Neprotejarea secţiunilor din codul sursă la generarea de excepţii;
- Conectarea eronată sau neconectarea la IED-uri;
- Apariţia unor bucle infinite în secvenţele de achiziţionare de informaţii sau evenimente;
- Netratarea tuturor condiţiilor care pot să apară în procesul de funcţionare a echipamentelor electrice, etc.

În concluzie, problemele care pot cauza proasta funcţionare a unui sistem de monitorizare şi control a unei staţii electrice pot fi de mai multe feluri, de la întârzierea transmisiei datelor, afişarea eronată a datelor, până la nefuncţionarea unei componente software sau chiar a întregului sistem.

În cazul în care staţia electrică nu are personal, astfel de erori pot produce pagube însemnate, ajungându-se la defecte fizice chiar şi în cazul în care sistemul a avut erori doar pe partea de monitorizare nu şi de control. În cazul în care sistemul include pe lângă partea de monitorizare şi parte de control, în urma erorilor din procesul de monitorizare, poate fi afectat şi controlul echipamentelor, deoarece acest tip de operaţiune este urmarea la informaţiile obţinute prin monitorizare.

Informaţiile obţinute cu ajutorul sistemului de monitorizare şi control sunt utile la dispeceratul energetic naţional (DEN) iar o întârziere de raportare a unui defect al echipamentului electric monitorizat poate afecta echilibrul energetic pe o anumita zonă. În consecinţă, domeniul energetic, fiind un domeniu critic, în special la noi în ţară dar nu numai, începe să resimtă vechimea echipamentelor electrice actuale. Deoarece este imposibilă o reînnoire rapidă a tuturor echipamentelor cu probleme, soluţia este de a le prelungi durata de viaţă (prin procedee tehnologice specifice şi dedicate fiecărui tip de echipament electric) şi de a le monitoriza parametrii principali pentru prevenirea efectelor defectării lor.

Importanţa unui sistem complex de monitorizare şi control al staţiilor electrice a fost conştientizată de marii operatori de transport şi distribuţie a energiei electrice şi cerinţele sunt ca aceste sisteme să funcţioneze fără erori/defecte.

3.5. Concluzii

În acest capitol este prezentată arhitectura generală a unui sistem de monitorizare şi control a unei staţii electrice. Sunt prezentate sisteme existente reprezentative, produse de fabricanţi cunoscuţi în domeniul energetic. Este prezentat sistemul de monitorizare şi control EMCSIT, utilizat ca studiu de caz în cadrul tezei.

Tendinţa internaţională şi recent naţională, este de a crea reţele inteligente (Smart Grids) ce includ sisteme de monitorizare şi control în domeniul energetic. Investiţiile sunt masive şi realizabile pe termen mediu şi lung.

Sunt identificate problemele ce pot apărea în dezvoltarea, funcţionarea şi operarea acestui sistem şi care pot cauza căderea lui.

Importanţa unui sistem complex de monitorizare şi control a unei staţii electrice a fost conştientizată de marii operatori de transport şi distribuţie a energiei electrice şi cerinţele sunt ca aceste sisteme să funcţioneze fără erori sau defecte.

4. Studiu de caz: soluţii pentru asigurarea calităţii sistemului de monitorizare şi control EMCSIT

4.1. Particularităţile sistemului

EMCSIT este un sistem complex de monitorizare on-line a unei staţii electrice [Urs8-3]. Arhitectura sistemului EMCSIT este de tip client-server.

Din punct de vedere hardware, sistemul de monitorizare este alcătuit din mai multe IED-uri (Intelligent Electronic Device) poziţionate în fiecare cabină de relee din cadrul staţiei electrice, care sunt conectate la senzori şi traductori montaţi pe echipamentele electrice din staţie şi transmit informaţiile achiziţionate de la acestea către serverele locale.

Sistemul de monitorizare, din punct de vedere informatic, include calculatoare (denumite în continuare servere) care sunt conectate la echipamentele de monitorizare (IED-uri) şi achiziţionează valorile parametrilor monitorizaţi, pe care le stochează într-o bază de date locală Oracle. Pe lângă aceste servere, există un server central care efectuează sincronizarea cu celelalte servere şi preia informaţiile pentru a fi accesate de aplicaţiile client EMCSIT din reţeaua locală.

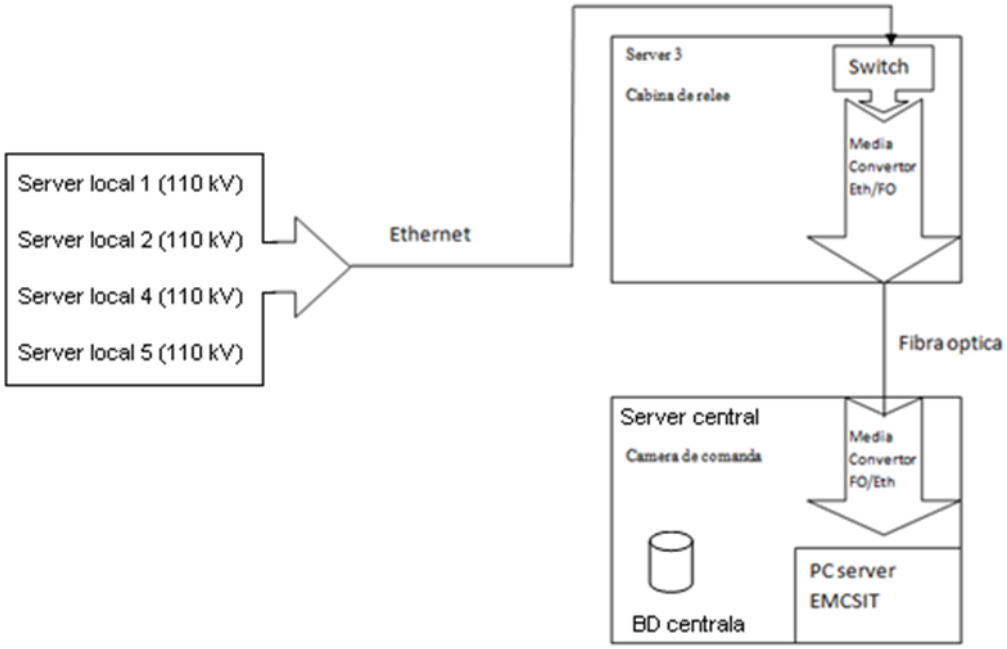

Fig. 4.1.1. Conexiunea serverelor locale cu serverul central

Deoarece un server local se conectează la mai multe IED-uri, va trebui să ruleze câte o instanţă de aplicaţie server pentru fiecare IED conectat. Pentru a preveni rularea a mai

multor ferestre de aplicaţii server, fiecare conectându-se la câte un IED, a fost dezvoltată o aplicaţie denumită „EMCSIT SuperServer”. Aceasta rulează câte o instanţă de aplicaţie server aferentă fiecărui IED, transparent pentru utilizator, utilizând mai multe thread-uri (fire de execuţie). În cazul în care există erori pentru achiziţia informaţiilor de la un IED, thread-ul respectiv va fi abandonat, urmând ca operaţiunea să se reia la următorul interval de achiziţie. În cadrul sistemului folosit ca studiu de caz, intervalul de achiziţie este de 1 minut. Astfel, la fiecare minut, aplicaţia EMCSIT SuperServer interoghează toate IED-urile care sunt conectate la serverul local şi achiziţionează informaţii privind parametrii monitorizaţi ai echipamentelor electrice.

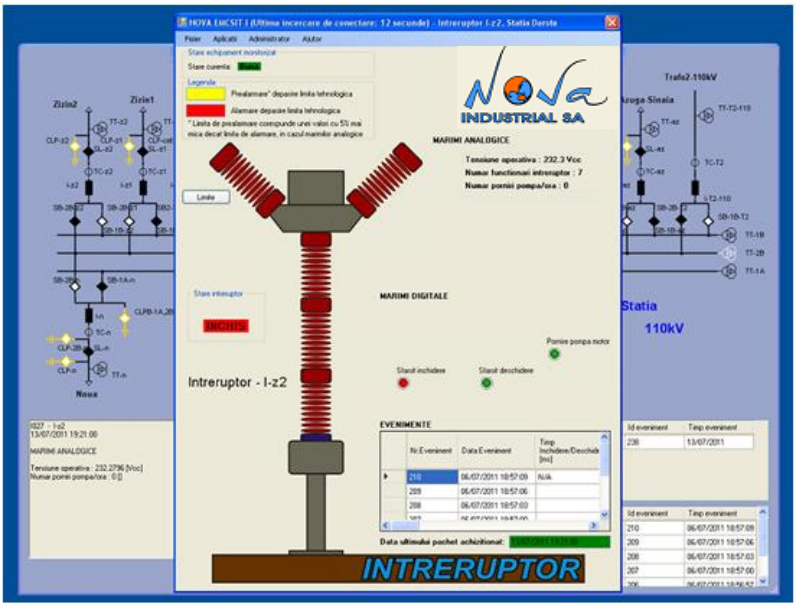

Fig. 4.1.2.Aplicaţia EMCSIT Staţie ce prezintă schema monofilară a staţiei şi integrarea aplicaţiei EMCSIT client pentru monitorizarea unui întreruptor

După achiziţionarea pachetului de date de la IED, informaţiile sunt prelucrate şi salvate în baza de date, instalată pe serverul central aferent câte unei substaţii din componenţa staţiei electrice. În cazul în care se produce un eveniment (închidere/deschidere întreruptor sau separator, supracurenţi, supratensiuni, descărcare, etc.) pentru un echipament electric monitorizat, se va deschide un alt thread pentru achiziţionarea, prelucrarea şi salvarea lui în baza de date, utilizând un modul software dedicat, EMCSIT Evenimente.

În perioada de dezvoltare a sistemului, au fost colectate informaţii privind numărul de defecte descoperite la nivelul fiecărei aplicaţii din componenţa sistemului. Pe baza acestor informaţii, a fost calculată rata de defectare a fiecărei componente software a sistemului.

Calitatea sistemului trebuie să fie asigurată la nivelul mai multor componente:

- componentele de achiziţie de date (IED-uri);
- componentele de transmisie a datelor (integritatea fizică a mediilor de transmisie şi asigurarea securităţii şi integrităţii datelor transmise);
- aplicaţiile server de prelucrare (aplicaţiile server EMCSIT trebuie să asigure achiziţia corectă a informaţiilor, prelucrarea acestora şi stocarea în baza de date locală);
- aplicaţiile client de vizualizare a parametrilor (aplicaţiile client EMCSIT trebuie să asigure corectitudinea achiziţiei datelor de la serverul EMCSIT şi promptitudinea achiziţiei).

Din experienţa avută cu sistemele de monitorizare şi control cu care am lucrat, am constatat că, în timp, pot apărea defectări atât la nivelul componentelor hardware cât şi a celor software.

Printre erorile hardware cu care am fost confruntat, se numără:

- defectarea unui microcontroller din diverse cauze, cum ar fi cele termice;
- defectarea unui releu; astfel, comenzile dispozitivului hardware nu ajung la echipamentul electric comandat;
- funcţionarea incorectă a unui traductor de curent, ceea ce duce la calcule eronate şi la comenzi greşite;
- trasee electronice întrerupte pe placă, etc.

Printre erorile software observate şi modalităţile în care pot fi depistate, menţionez:

- erori datorate ne-securizării accesului la interfaţa cu utilizatorul;
- netratarea tuturor posibilităţilor ce pot apărea în exploatare, în calcule, ce pot duce la rezultate eronate. Aceasta se verifică folosind testarea structurală pentru a detecta căi absente în graful program;
- netratarea situaţiei în care traductorii de curent sau temperatură sunt neconectaţi. Aceasta se verifică prin testarea pentru a descoperi intrări nevalide;
- netratarea împărţirilor la 0. Sunt verificate căile absente în graful program prin testare structurală/funcţională;
- autoscalarea temporizată greşit a mărimilor ce duce la intrarea în bucle infinite;

- proiectarea şi implementarea software greşită, astfel încât la anumite erori să se depăşească memoria alocată sau stiva şi să se întrerupă funcţionarea normală. Se verifica prin testarea fiabilităţii sistemului;
- pornirea dispozitivului cu stările iniţiale ale comenzilor specificate incorect, ce poate duce la alarmări false, fără a fi îndeplinite condiţiile de alarmare (semnalizare sau declanşare) din cauza variabilelor neiniţializate sau iniţializate incorect. Aceasta se verifică prin testarea software folosind simulatoare ale IED-urilor;
- semnalizare eronată, conducând la alertarea falsă a personalului sau chiar la deconectarea din SEN (sistemul energetic naţional) a unui echipament electric (transformator/autotransformator) funcţional şi posibilitatea dezechilibrării SEN pe zona respectivă;

Este propusă pentru viitor, ca soluţie pentru remedierea acestor tipuri de erori/defecte, implementarea unui watchdog hardware (mecanism de reiniţializare a IED-ului în cazul blocării), tratarea cât mai multor excepţii posibile în software, validarea internă a rezultatelor înainte de a trimite comenzi către echipamentul electric monitorizat.

4.2. Dezvoltarea unui simulator de echipamente de monitorizare şi control (IED-uri) pentru testarea sistemului

Dezvoltarea unui sistem de monitorizare şi control a unei staţii electrice include atât componente software cât şi hardware şi necesită un efort financiar şi uman uriaş.

Este necesar ca un astfel de sistem să fie cât mai fiabil. Pentru aceasta se efectuează teste independente pentru componentele software şi apoi teste de acceptanţă în staţie, având toate subsistemele conectate şi funcţionale.

În continuare voi prezenta testele efectuate în cadrul proiectului EMCSIT, prezentat ca studiu de caz. Aceste teste au fost făcute înaintea dezvoltării simulatorului de IED-uri şi prin urmare nu au beneficiat de nici unul din avantajele utilizării acestuia.

Testarea aplicaţiilor EMCSIT s-a efectuat în mai multe etape, înainte de instalarea în cadrul staţiei electrice:

- mai întâi a fost testată funcţionarea fiecărui tip de IED împreună cu aplicaţia EMCSIT Server corespunzătoare. Aceasta testare a presupus conectarea pe interfaţa serială a câte unui tip de IED şi simularea, utilizând diverse metode electrotehnice, a unor valori pentru parametrii monitorizaţi. Valorile folosite prin această metodă de

simulare nu au corespuns valorilor reale obţinute în mediul de lucru datorită imposibilităţii simulării condiţiilor de lucru reale în staţia electrică.

- au fost testate modulele EMCSIT Client, Grafice Evenimente şi Istoric împreună cu aplicaţia EMCSIT Server;
- a fost testată aplicaţia EMCSIT Staţie;

A urmat instalarea aplicaţiilor software în cadrul staţiei electrice şi au fost efectuate teste de acceptanţă, cu unele echipamente electrice neconectate la SEN (Sistemul Energetic Naţional). Datorită structurii staţiei electrice, funcţionalităţii şi disponibilităţii acesteia, nu s-au putut efectua testele de acceptanţă cu toate echipamentele electrice neconectate, cum era recomandat. După ce aceste teste au fost trecute, echipamentele electrice (ce au putut fi deconectate) au fost repuse în operare pentru ca testarea întregului sistem de monitorizare şi control să aibă loc în condiţii reale de exploatare.

Toate aceste teste au avut drept scop obţinerea unui produs software fiabil, care să poată fi utilizat în bune condiţii de către beneficiar. Beneficiarul direct este personalul operativ din staţie, care utilizează acest software şi trimite informaţiile obţinute către dispeceratul zonal şi cel naţional (DEN).

Dificultăţile şi insuficienţa procesului de testare descris mai sus sunt legate de următoarele aspecte:

- În absenţa echipamentelor electrice monitorizate de IED-uri şi a unor simulatoare ale IED-urilor, aplicaţiile server au fost testate insuficient;
- Accesul şi perioada petrecută pentru teste în cadrul unei staţii electrice necesită aprobări speciale şi este limitată. Din acest motiv, nici perioada de testare de acceptanţă nu este suficientă pentru descoperirea unui procent cât mai mare din posibilele defecte.

De aceea, am considerat necesară dezvoltarea unui simulator software pentru IED-uri, ce aduce următoarele avantaje:

- Posibilitatea testării mult mai complete a componentelor software de tip server;
- Posibilitatea simulării conectării mai multor IED-uri la o aplicaţie server;
- Testarea aplicaţiilor server cu valori reale, conform limitelor specificate de specialiştii tehnologi pentru fiecare tip de echipament electric.
- Executarea testelor software înainte de existenţa IED-urilor;

- Posibilitatea descoperirii şi demonstrării proastei funcţionării a unui IED prin compararea rezultatelor obţinute utilizând aplicaţia server împreună cu un IED şi utilizând aplicaţia server împreună cu simulatorul software;
- Scurtarea perioadei de teste de acceptanţă în staţia electrică;
- Obţinerea unui software mult mai fiabil;
- Reducerea costurilor legate de dezvoltarea şi mentenanţa sistemului prin realizarea componentelor software în paralel cu procesul de producţie a IED-urilor.

Acest simulator de IED-uri transmite pe interfaţa serială pachete de date ce pot fi preluate de către aplicaţiile tip server EMCSIT.

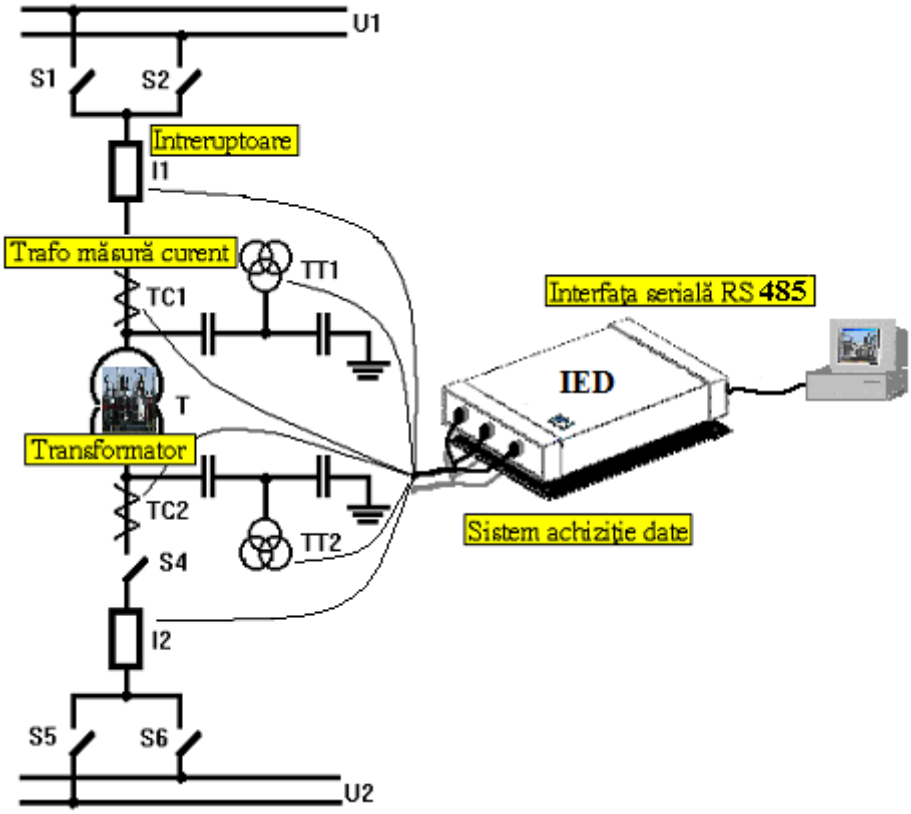

Fig. 4.2.1. Ansamblul echipament electric – IED – server local

Utilizând aceste pachete de date, sunt testate:

- achiziţia informaţiilor de pe interfaţa seriala;
- corectitudinea calculelor privind parametrii monitorizaţi;
- salvarea corectă a rezultatelor în baza de date;

În acest fel, se pot descoperi şi elimina defecte înainte de instalarea în staţie.

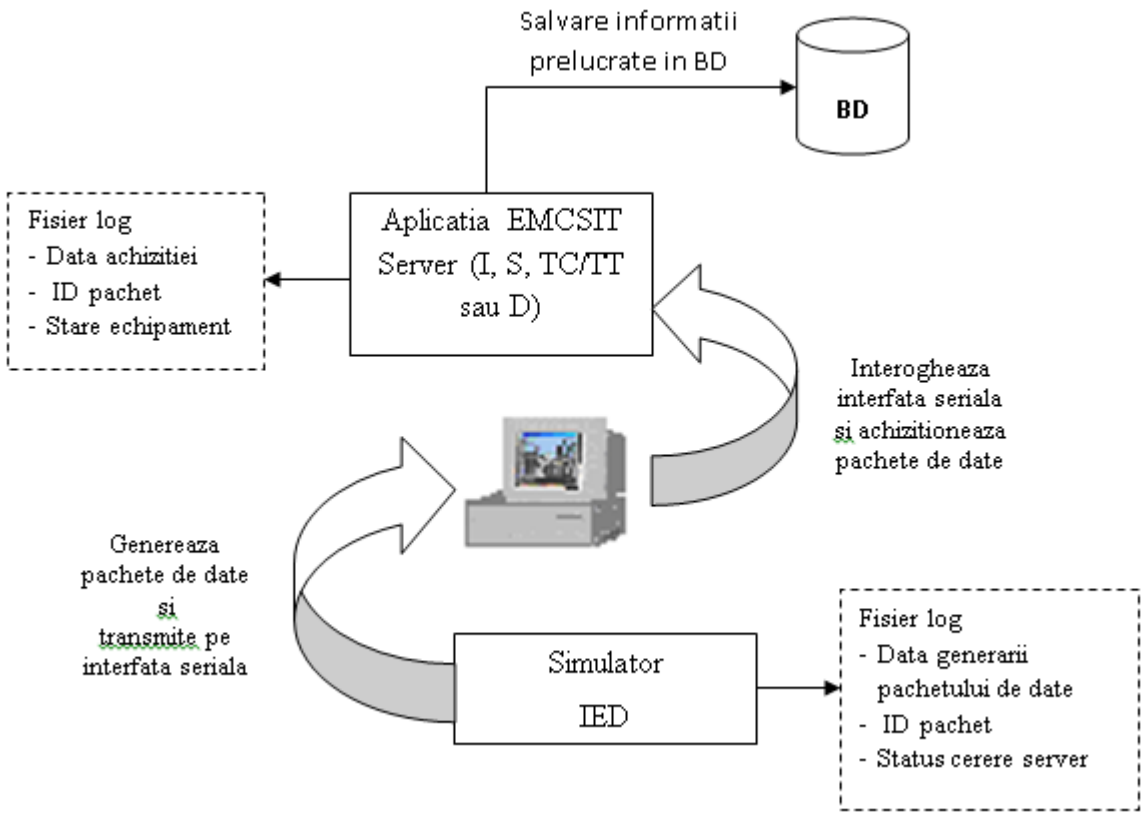

Fig. 4.2.2. Ansamblul simulator IED – aplicaţie server EMCSIT

Pachetele de date transmise de către simulator sunt generate pe baza unor mărimi de intrare specifice echipamentelor electrice conectate la IED. Mărimile de intrare diferă de la un tip de IED la altul.

Totodată, protocolul de comunicaţie între IED şi server nu este acelaşi pentru toate tipurile de IED. De aceea, înainte de pornirea simulatorului, este necesară o etapă de configurare în care tester-ul:

- alege tipul de IED ce va fi simulat;
- precizează numărul mărimilor monitorizate de IED;
- specifică domeniul de valori al fiecărei mărimi monitorizate şi limitele de alarmare;

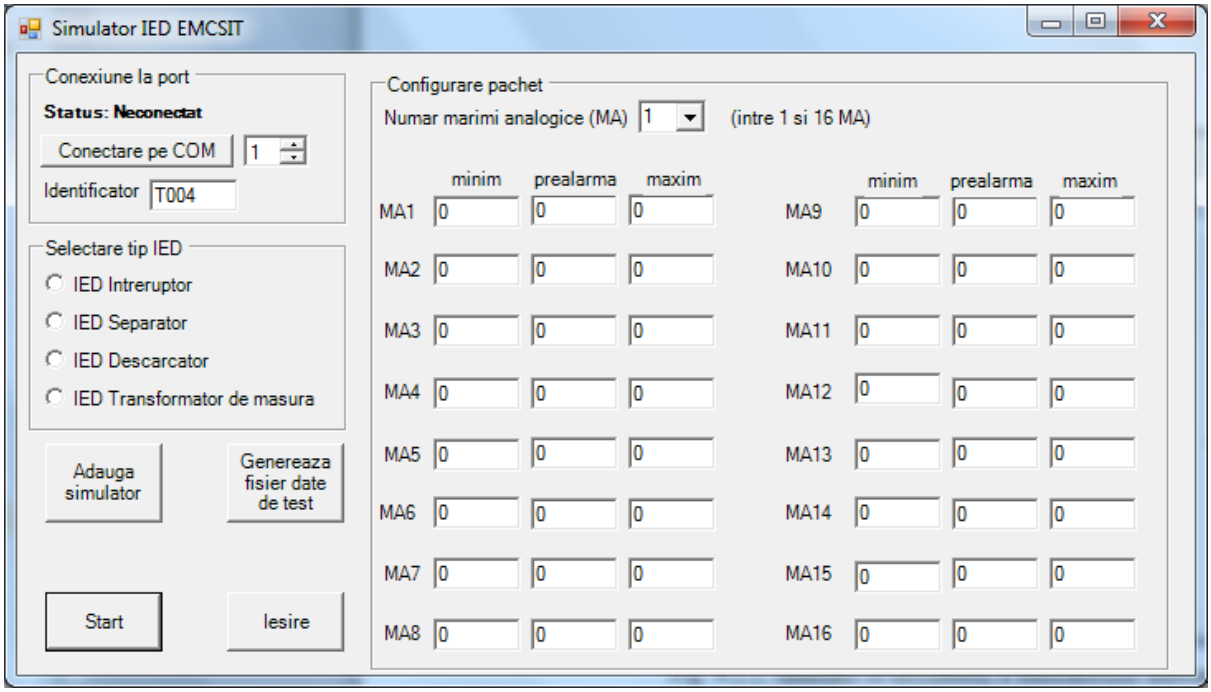

Fig. 4.2.3. Fereastra de configurare a simulatorului de IED-uri

În fereastra de configurare a simulatorului software, trebuie mai întâi selectat portul serial pe care se va comunica cu aplicaţia EMCSIT server. Apoi se va selecta tipul de IED care trebuie să fie simulat.

În momentul în care este selectat numărul mărimilor analogice (parametrii monitorizaţi), vor fi editabile căsuţele text pentru valorile - minim, prealarmă şi maxim doar pentru acel număr de mărimi specificat. Trebuie completate toate cele trei valori pentru fiecare mărime analogică.

Pasul următor îl reprezintă generarea fişierului cu date de test utilizând metoda Pairwise testing.

Simularea poate fi pornită pentru IED-ul care a fost configurat sau se poate adăuga alt IED, datele pentru IED-ul curent fiind salvate într-un fişier text de configurare.

Simulatorul are posibilitatea adăugării mai multor IED-uri virtuale datorită implementării protocolului de comunicare între IED-uri – Daisy Chain.

IED-urile sunt conectate la un calculator server într-o configuraţie Daisy-Chain prin conectarea fiecărui IED la alt IED, nu prin conectarea individuală, a fiecărui IED direct la server. Numai ultimul IED din această înlănţuire se conectează direct la server. Fiecare IED are un identificator unic, comunicarea făcându-se pe aceeaşi magistrală în funcţie de valoarea acestuia.

Astfel, utilizând simulatorul pot fi simulate mai multe IED-uri pe acelaşi port serial al calculatorului server cu condiţia ca identificatorul fiecărui IED să fie diferit. Nu există restricţii privind ordinea adăugării IED-urilor în simulator şi nu trebuie respectată ordinea conectării fizice a acestora.

În momentul în care se porneşte simulatorul, la intervale predefinite de timp (valoarea iniţială este de 5 secunde), vor fi generate pachete de date şi trimise pe interfaţa serială pentru achiziţia acestora de către server.

Simulatorul generează fişiere log ce conţin informaţii privind data generării pachetului de date, id-ul acestuia şi dacă a fost achiziţionat cu succes sau nu de către aplicaţia server.

Exemplu de fişier log generat de simulator:

- 8-10-2011_12-10-37-755 Primit solicitare de la server pentru pachetul cu ID - 1
- 8-10-2011_12-10-47-489 Primit solicitare de la server pentru pachetul cu ID - 2
- 8-10-2011_12-10-57-629 Primit solicitare de la server pentru pachetul cu ID - 3
- 8-10-2011_12-11-7-770 Primit solicitare de la server pentru pachetul cu ID - 4
- 8-10-2011_12-11-17-910 Primit solicitare de la server pentru pachetul cu ID – 5

În cazul de faţă, pachetele de date generate de către simulator au fost achiziţionate cu succes de către aplicaţia server.

Atunci când se simulează un singur IED, pentru testare va fi utilizată aplicaţia EMCSIT Server corespunzătoare acelui tip de IED.

Dacă au fost adăugate mai multe IED-uri, atunci va trebui utilizată pentru testare aplicaţia EMCSIT SuperServer care are posibilitatea rulării în paralel a mai multor instanţe ale aplicaţiilor EMCSIT Server pentru tipurile de IED-uri simulate.

În acest moment, simulatorul are implementat un protocol de comunicaţie proprietar.

Există posibilitatea adaptării acestui simulator prin:

- Adăugarea posibilităţii conectării prin interfaţa de reţea;
- Implementarea altor protocoale de comunicaţie, precum cel definit de standardul IEC61850.
- Simularea trimiterii de comenzi către echipamentele electrice;

4.2.1. Simularea unui IED pentru un întreruptor

Întreruptoarele electrice sunt aparate cu ajutorul cărora se realizează operaţiile necesare de conectare şi de deconectare atât în condiţii normale de lucru, cât şi în condiţii de avarie.

Ţinându-se cont că aceste aparate trebuie să facă faţă la curenţi cu intensităţi foarte mari care pot apărea frecvent în reţelele electrice, ele se caracterizează, în afara parametrilor nominali comuni tuturor aparatelor electrice de comutaţie (curent nominal, tensiune nominală, frecvenţă de lucru, curent limită dinamic, curent limită termic), şi printr-o altă mărime nominală specifică, ce defineşte proprietăţile de comutaţie şi care se numeşte capacitate nominală de rupere în scurtcircuit. Ea reprezintă cel mai mare curent de scurtcircuit (valoare efectivă) pe care un întreruptor este capabil sa-l întrerupă în condiţii de utilizare şi funcţionare prescrise.

Pentru a testa aplicaţiile server EMCSIT s-au aplicat praguri limită pentru datele achiziţionate: tensiune operativă, număr porniri pompă/oră, astfel încât să fie simulată starea bună, starea de prealarmare şi starea de alarmare pentru echipamentul electric, conform informaţiilor primite de la tehnologi.

Pentru un întreruptor se poate testa aplicaţia server utilizând 2 mărimi analogice:

- Mărimea analogică 1 (MA1): Tensiunea operativă;
- Mărimea analogică 2 (MA2): Număr porniri pompă/oră;

Limitele minime şi maxime ce definesc intervalul de valori pentru fiecare mărime analogică, sunt următoarele:

- Pentru MA1: valoarea minimă este 0 Vcc (volţi curent continuu) iar valoarea maximă este de 400 Vcc;
- Pentru MA2: nu sunt emise alarme, dar se poate considera ca valoare minimă un număr de 0 funcţionări/oră iar valoarea maximă este de 255 funcţionări/oră;

Limitele de prealarmare pentru aceste mărimi şi care sunt transmise ca date de intrare pentru simulatorul de IED-uri sunt următoarele:

- Pentru MA1: 260 Vcc;

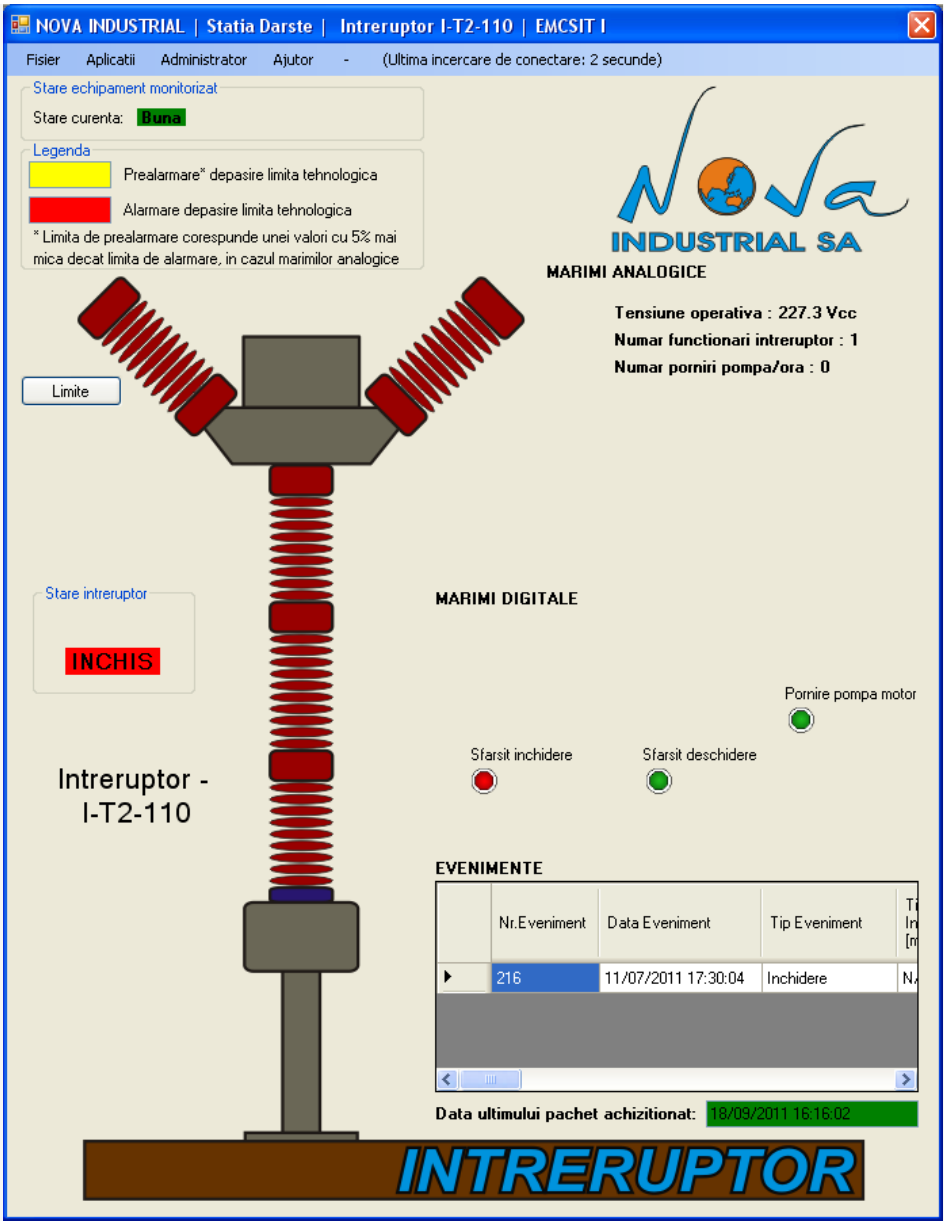

Fig. 4.2.1.1. Aplicaţia server pentru întreruptor.

4.2.2. Simularea unui IED pentru un separator

Normele de protecţia muncii cer ca lucrările de întreţinere şi reparaţii în instalaţiile de înaltă tensiune să se execute numai după ce circuitul în care se lucrează a fost deconectat şi izolat vizibil de restul instalaţiei. De asemenea, în instalaţiile electrice sunt necesare uneori manevre de conectare sau de deconectare a unor circuite fără curent (schimbarea sau separarea barelor, trecerea de la un generator sau transformator la altul etc.), fiind necesare aparate de conectare simple, cu construcţie robustă, cu manevrare uşoară şi poziţie uşor vizibilă. Pentru aceste scopuri se folosesc separatoarele de înalta tensiune.

Separatoarele sunt aparate de comutaţie destinate conectării şi deconectării circuitelor sub tensiune, însă fără sarcină. Ele pot întrerupe şi curenţi de intensitate redusă, cum sunt curenţii de magnetizare (de mers în gol) ai transformatoarelor de putere mică sau curenţii nominali ai transformatoarelor de putere foarte mică.

Separatoarele de înaltă tensiune se realizează în numeroase tipodimensiuni, care se deosebesc prin parametrii nominali (tensiuni între 35 şi 750 kV, curenţi între 200 şi 6000 A), prin numărul de poli şi varianta constructivă.

Într-o instalaţie energetică separatorul permite efectuarea transferului de energie între sistemele de bare colectoare la plecări sau sosiri. Cu ajutorul separatoarelor se poate face cuplarea unei linii la unul din sistemele de bare colectoare.

Pentru a testa aplicaţiile server EMCSIT s-au aplicat praguri limită pentru datele achiziţionate: număr funcţionări, astfel încât să fie simulată starea bună, starea de prealarmare şi starea de alarmare pentru echipamentul electric, conform informaţiilor primite de la tehnologi.

Pentru un separator se poate testa aplicaţia server utilizând o singură mărime analogică:

- Mărimea analogică 1 (MA1): număr funcţionări.

Limitele minimă şi maximă pentru aceasta mărime sunt următoarele:

- valoarea minimă: 0 funcţionari, valoarea maximă: 255 funcţionări; nu sunt emise alarme.

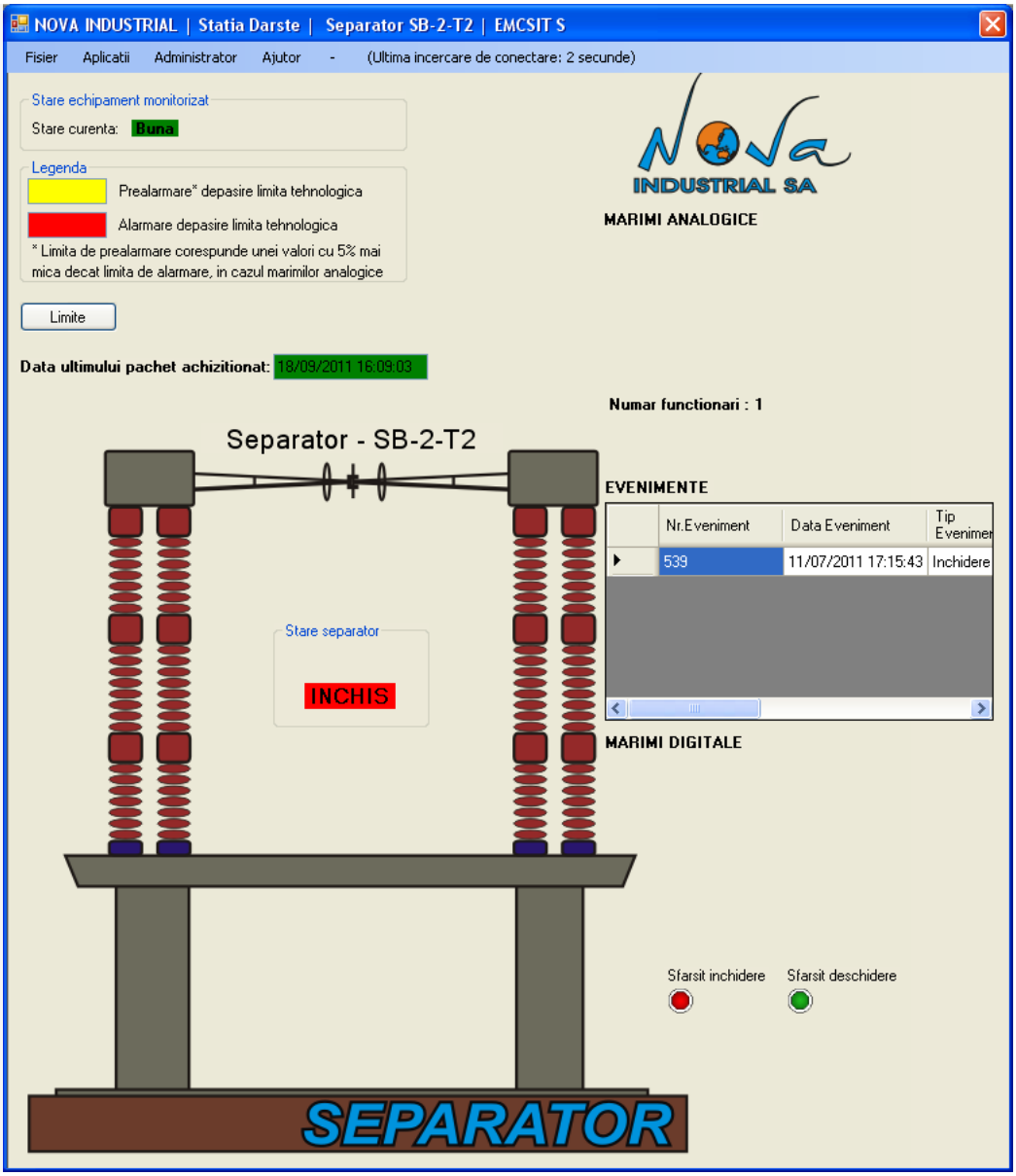

Fig. 4.2.2.1. Aplicaţia server pentru un separator.

4.2.3. Simularea unui IED pentru un descărcător

Descărcătoarele sunt aparate de protecţie care pe lângă funcţia principală de limitare a supratensiunilor sunt capabile să reducă curentul de însoţire la valori pentru care spaţiul disruptiv devine izolant, fiind prevăzute cu dispozitive speciale de stingere a arcului electric, imediat ce tensiunea a revenit la valori nepericuloase pentru instalaţie.

Rolul funcţional al descărcătorului electric este de a limita supratensiunile atmosferice şi de comutaţie într-o instalaţie electrică. Descărcătorul se montează la intrarea în staţiile electrice între faze şi pământ şi în punctele în care linia îşi modifica impedanţa caracteristică.

Pentru a testa aplicaţiile server EMCSIT s-au aplicat praguri limită pentru datele achiziţionate: curenţi armonică 1 şi armonică 3, curent de dezechilibru astfel încât să fie simulată starea bună, starea de prealarmare şi starea de alarmare pentru echipamentul electric, conform informaţiilor primite de la tehnologi.

Pentru un descărcător se poate testa aplicaţia server utilizând 10 mărimi analogice:

- Mărimea analogică 1 (MA1): Curent total faza R;
- Mărimea analogică 2 (MA2): Curent armonică a 3-a faza R;
- Mărimea analogică 3 (MA3): Curent total faza S;
- Mărimea analogică 4 (MA4): Curent armonică a 3-a faza S;
- Mărimea analogică 5 (MA5): Curent total faza T;
- Mărimea analogică 6 (MA6): Curent armonică a 3-a faza T;
- Mărimea analogică 7 (MA7): Contor faza R;
- Mărimea analogică 8 (MA8): Contor faza S;
- Mărimea analogică 9 (MA9): Contor faza T;
- Mărimea analogică 10 (MA10): Curent total dezechilibru;

Limitele minime şi maxime care definesc intervalul de valori pentru fiecare mărime analogică, sunt următoarele:

- Pentru MA1, MA3 şi MA5: valoarea minimă este de 0 microamperi iar valoarea maximă este de 100000 microamperi;
- Pentru MA2, MA4 şi MA6: valoarea minimă este de 0 microamperi iar valoarea maximă este de 1000 microamperi;
- Pentru MA7, MA8 şi MA9: nu sunt emise alarme, dar se poate considera că valoare minimă un număr de 0 funcţionări iar valoarea maximă este de 255 funcţionări;
- Pentru MA10: valoarea minimă este de 0 microamperi iar valoarea maximă este de 100000 microamperi;

Limitele de prealarmare pentru aceste mărimi şi care sunt transmise ca date de intrare pentru simulator sunt următoarele:

- Pentru MA1, MA3 şi MA5: 1000 microamperi;
- Pentru MA2, MA4 şi MA6: 100 microamperi;
- Pentru MA10: 200 microamperi;

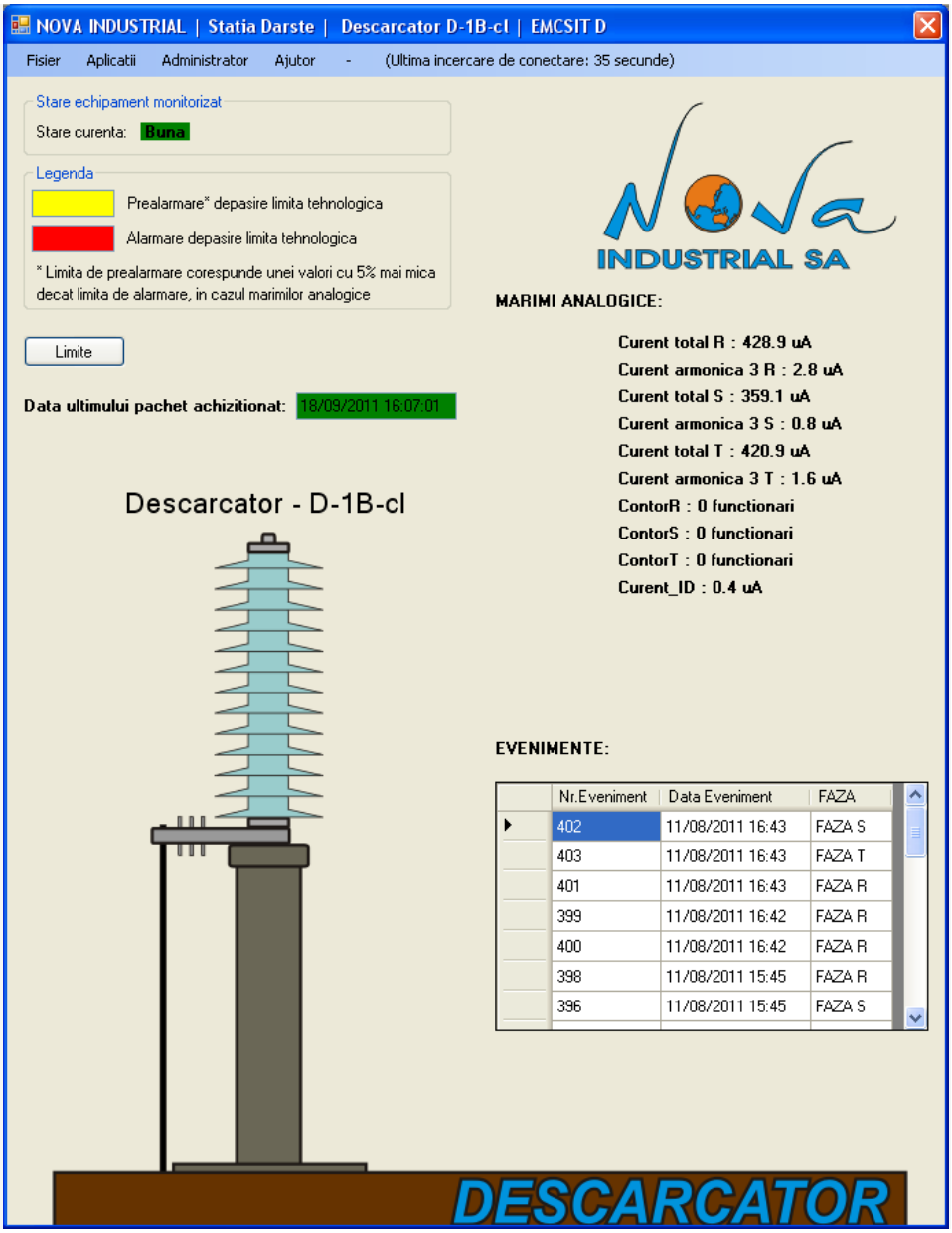

Fig. 4.2.3.1. Aplicaţia server pentru descărcător

4.2.4. Simularea unui IED pentru un transformator de tensiune/curent

Transformatoarele sunt aparate electroenergetice statice, care transformă parametrii energiei electrice, tensiunea respectiv curentul, reducând valoarea acestora de un anumit număr de ori.

Pentru o exploatare în bune condiţii a sistemului energetic sunt necesare aparate de măsură a mărimilor electrice: curent, tensiune, putere, frecvenţă, etc., aparate de protecţie în vederea asigurării funcţionarii corecte într-un regim anormal sau de avarie în instalaţie, aparate de reglare automată, care realizează reglarea tensiunii, a frecvenţei, etc.

În acest sens se impune ca soluţie: utilizarea transformatoarelor de măsură, destinate transformării valorilor curentului şi a tensiunii din circuitele primare în valori convenabile pentru circuitele secundare (1A, 2A sau 5A respectiv 100V).

Pentru a testa aplicaţiile server EMCSIT s-au aplicat praguri limită pentru datele achiziţionate: curenţi şi tensiuni astfel încât să fie simulată starea bună, starea de prealarmare şi starea de alarmare pentru echipamentul electric, conform informaţiilor primite de la tehnologi.

Pentru un transformator de curent/tensiune se poate testa aplicaţia server utilizând 6 mărimi analogice:

- Mărimea analogică 1 (MA1): Curent faza R;
- Mărimea analogică 2 (MA2): Curent faza S;
- Mărimea analogică 3 (MA3): Curent faza T;
- Mărimea analogică 4 (MA4): Tensiune faza R;
- Mărimea analogică 5 (MA5): Tensiune faza S;
- Mărimea analogică 6 (MA6): Tensiune faza T;

Limitele minime şi maxime ce definesc intervalul de valori pentru fiecare mărime analogică, sunt următoarele:

- Pentru MA1, MA2 şi MA3: valoarea minimă este de 0 Amperi iar valoarea maximă este de 2500 Amperi;
- Pentru MA4, MA5 şi MA6: valoarea minimă este de 0 kiloVolţi iar valoarea maximă este de 500 kiloVolţi;

Limitele de prealarmare impuse de tehnologi pentru aceste mărimi şi care sunt transmise ca date de intrare pentru simulator sunt următoarele:

- Pentru MA1, MA2 şi MA3: 1920 Amperi;
- Pentru MA4, MA5 şi MA6: 255 kiloVolţi;

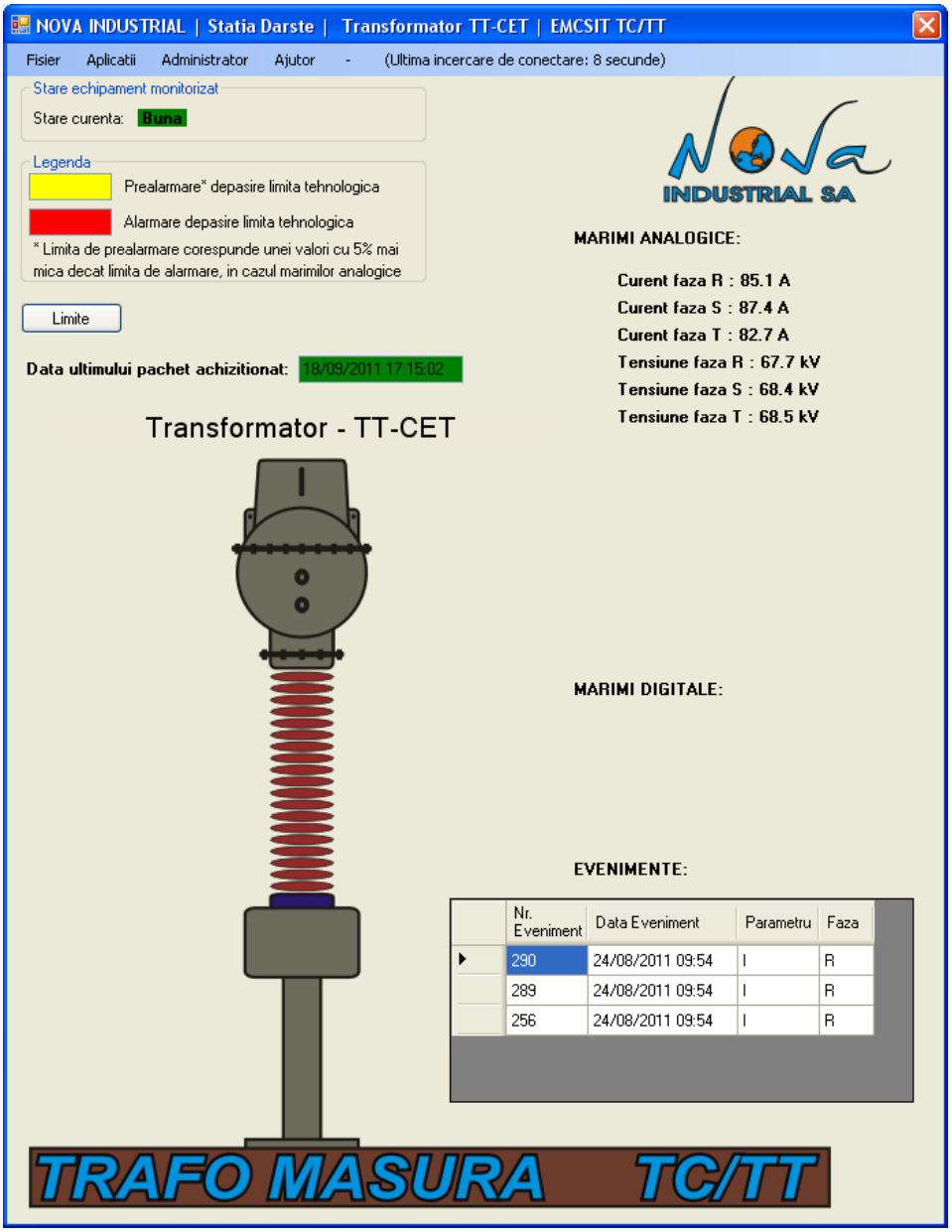

Fig. 4.2.4.1. Aplicaţia server pentru un transformator de curent/tensiune

4.3. Generarea cazurilor de test– Metoda Pairwise testing

Generarea datelor de test prin analiza domeniilor variabilelor de intrare conduce la foarte multe cazuri de test: vectorii de test reprezintă combinaţii ale valorilor selectate pentru variabilele de intrare. Generarea ieşirii aşteptate pentru un vector de test este relativ simplă: proiectantul testului calculează ieşirea aşteptată analizând specificaţia programului.

Generarea datelor de test prin analiza domeniilor variabilelor de ieşire se potriveşte situaţiei când numărul de cazuri de test obţinute este mai mic şi nu este necesar să se

considere diferite combinaţii ale valorilor variabilelor de ieşire. Generarea unui vector de intrare necesar pentru a produce o valoare de ieşire necesită analiza specificaţiei în direcţia inversă.

Pentru testarea aplicaţiilor de tip server, simulatorul generează date de test pornind de la specificaţiile mărimilor monitorizate: valoarea minimă, valoarea maximă şi valoarea de prealarmare. Acestea sunt valori întregi. Considerând şi valorile la frontiera domeniului, valoare minimă -1/+1, valoare maximă -1/+1 precum şi valoare de prealarmare -1/+1, rezultă 9 valori posibile care trebuie generate de simulator pentru o mărime monitorizată.

In cazul în care trebuie simulat un IED care monitorizează N mărimi, varianta de testare cea mai costisitoare este de a genera toate combinaţiile posibile între cele 9 valori de test ale celor N mărimi, adică 9^N.

Numărul de cazuri de test obținut prin considerarea tuturor combinațiilor posibile între valorile variabilelor de intrare poate fi redus prin diferite metode.

Pairwise testing este una dintre aceste metode [Swq9-5]. Utilizarea acestei metode asigură faptul că fiecare combinaţie posibilă dintre valorile selectate pentru fiecare pereche de variabile de intrare este acoperită prin cel puţin un test.

Studii empirice au arătat că metoda poate conduce la detectarea a aproximativ 70% din defectele existente în software.

Pairwise testing este, de asemenea, menţionată ca testarea all-pair/two-way. Se poate, de asemenea, utiliza three-way/four-way testing pentru a acoperi toate combinaţiile de valori de trei sau patru variabile. Dacă toate combinaţiile de valori de mai multe variabile (de exemplu 2, 3, 4, ...) sunt luate în considerare, numărul cazurilor de test creşte.

Rezultate empirice pentru dispozitivele medicale şi sisteme distribuite de baze de date arată că folosirea Pairwise testing ar detecta mai mult de 90% din defecte, în timp ce four-way testing ar detecta 100% din defecte. Pentru browsere web şi aplicaţii server, Pairwise testing ar detecta aproximativ 70% din defecte, în timp ce six-way testing ar detecta 100% din defecte.

Tabel 4.3.1. Valorile de test pentru 6 mărimi analogice, fiecare cu 6 valori posibile (excluzând valorile pentru prealarmare): valoare minimă – 1, valoare minimă, valoare minimă + 1, valoare maximă – 1, valoare maximă, valoarea maximă + 1:

Parametrul 0 (mărime analogică MA1): -1, 0, 1, 399, 400, 401
Parametrul 1 (mărime analogică MA2):-1, 0, 1, 399, 400, 401

Parametrul 2 (mărime analogică MA3):-1, 0, 1, 399, 400, 401
Parametrul 3 (mărime analogică MA4):-1, 0, 1, 299, 300, 301
Parametrul 4 (mărime analogică MA5):-1, 0, 1, 299, 300, 301
Parametrul 5 (mărime analogică MA6):-1, 0, 1, 299, 300, 301

Utilizând metoda Pairwise testing se obţin 540 de cazuri de test, în loc de 6^6 (= 46656) cazuri, rezultate din considerarea tuturor combinaţiilor posibile. Un exemplu de cazuri de test obţinute prin această metodă este prezentat în anexa 5.

Având în vedere că a fost demonstrat statistic faptul că prin utilizarea Pairwise testing se descoperă aproximativ 70% din defecte dintr-un software, rezultă marele avantaj al utilizării acestei metode: economisirea timpului şi a costului testării [Swq9-5].

Simulatorul pe care l-am dezvoltat generează cazuri de test prin metoda Pairwise testing, pe care le transmite aplicaţiei server. Pot fi modificate atât intervalul de timp la care sunt transmise pachetele de date test cât şi numărul de mărimi analogice, simulatorul putând genera pachete de date cu maxim 16 mărimi analogice.

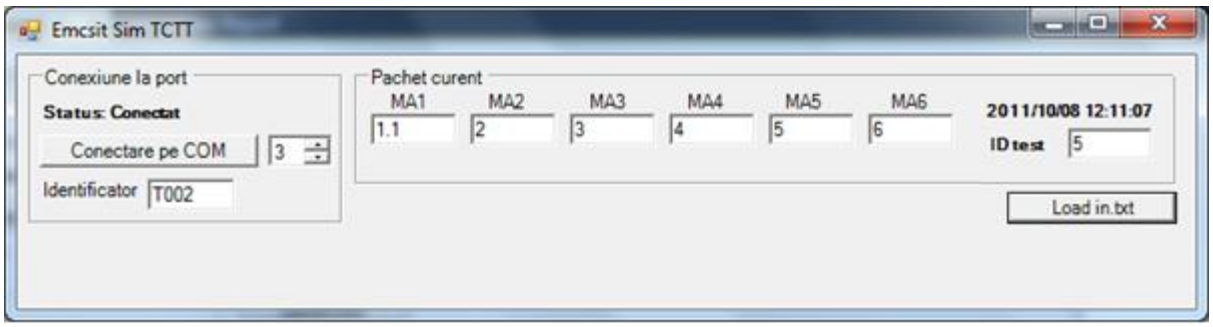

Fig. 4.3.1. Exemplu de utilizare a simulatorului de IED pentru un transformator de curent/tensiune

Aplicaţia server interoghează la intervale predefinite de timp (momentan intervalul este de 10 secunde) interfaţa serială a serverului şi achiziţionează pachetul de date ce conţine valorile mărimilor analogice monitorizate. Aceste informaţii sunt prelucrate şi afişate în cadrul interfeţei grafice cu utilizatorul. În cazul în care vreuna din valori depăşeşte limita tehnologică, este generată o avertizare.

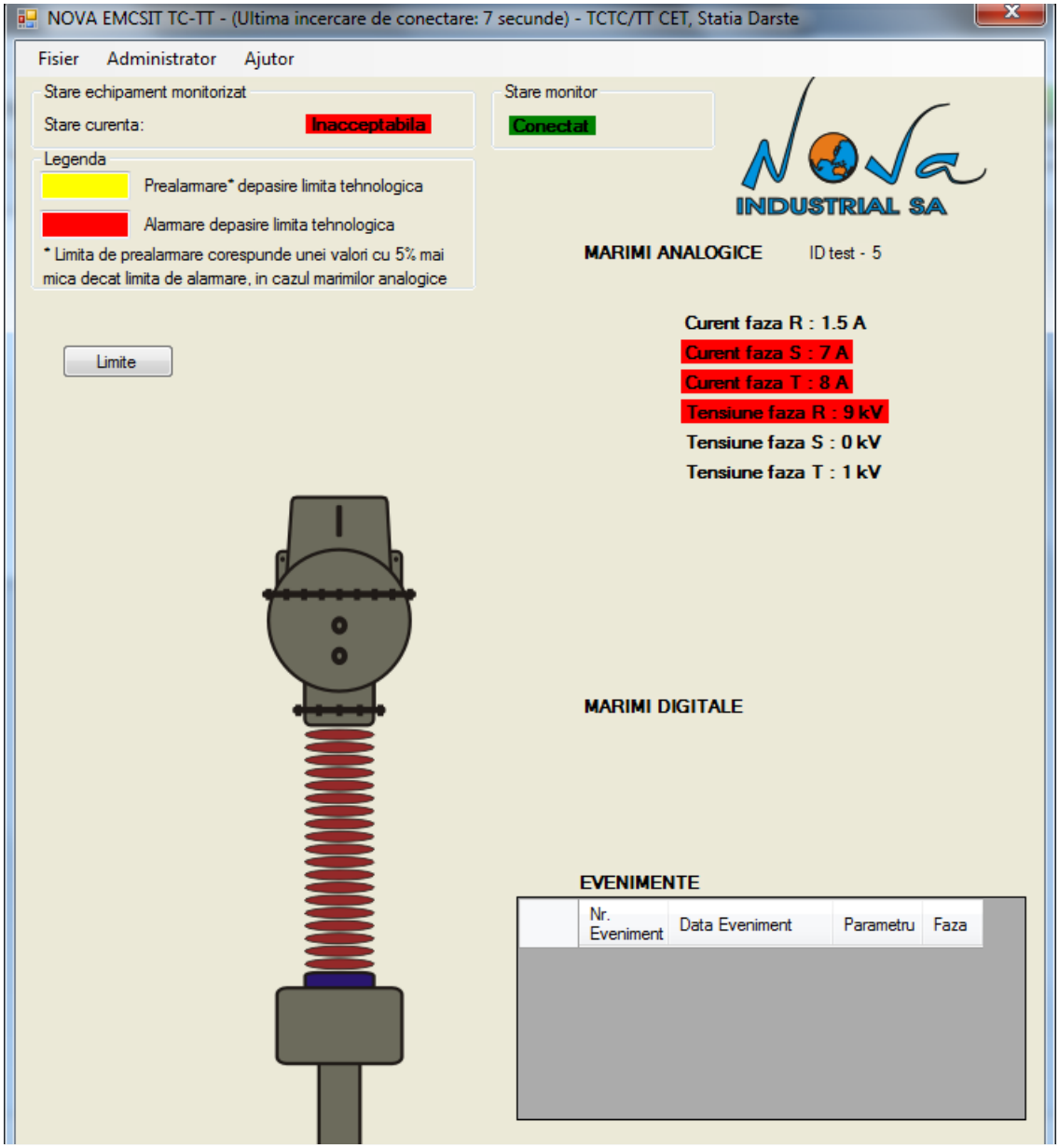

Fig. 4.3.2. Aplicaţia server EMCSIT pentru transformator de măsură curent/tensiune

Aplicaţia server generează un fişier log care conţine informaţii privind:

- Data pachetului de date achiziţionat;
- Valorile afişate;
- Starea echipamentului monitorizat;
- ID-ul pachetului de date conform cazului de studiu generat prin metoda Pairwise testing;

Utilizând fişierele log generate atât de simulator cât şi de aplicaţia server, se pot face corelări pentru a observa dacă au fost achiziţionate toate datele de test şi starea

echipamentului pentru acele valori, astfel putându-se verifica uşor corectitudinea calculelor efectuate în cadrul aplicaţiei server.

4.4. Asigurarea securităţii sistemului EMCSIT

Securitatea sistemului de monitorizare şi control EMCSIT trebuie asigurată pentru:

- Serverele locale şi serverul central;
- Transmisia de date între IED-uri şi serverele locale, între serverele locale şi cel central precum şi între serverul central şi clienţi;
- Accesul la aplicaţia EMCSIT Staţie, instalată pe calculatorul client din camera de comandă;

Actualele tehnici de securitate, implementate deja în cadrul sistemului de monitorizare EMCSIT, includ:

- autentificare prin utilizator/parolă pentru accesul la aplicaţia EMCSIT Staţie;
- asignarea unică de adrese IP pentru calculatoarele care fac parte din sistem;
- utilizarea de chei hardware HASP pentru asigurarea securităţii accesului la aplicaţia EMCSIT Staţie;
- implementarea algoritmului CRC pentru transmisia pachetelor de date între IED-uri şi serverele locale.

Pentru asigurarea securităţii accesului la serverele locale şi serverul central, aceste calculatoare au fost protejate prin metoda autentificării tip utilizator/parolă.

Protocolul de comunicaţie utilizat între serverele locale şi serverul central este TCP/IP iar securitatea transmisiei datelor este cea implementată şi inclusă în acest protocol.

Pentru moment, sistemul de monitorizare este accesibil doar în cadrul reţelei locale, prin intermediul aplicaţiei software EMCSIT Staţie instalată pe calculatorul client din camera de comandă.

Se intenţionează ca pe viitor să se configureze reţeaua pentru a fi utilizat protocolul IPv6.

4.5. Implementarea unor tehnici de toleranţă la defecte

4.5.1. Folosirea de Blocuri de Recuperare

Blocurile de recuperare pot fi utilizate în cadrul sistemului de monitorizare şi control EMCSIT în acele module sau secţiuni unde nu este necesară o viteză de lucru foarte ridicată.

Tehnica folosirii blocurilor de recuperare este utilizată în cadrul proiectului EMCSIT la achiziţia de evenimente (acţionări ale diverselor echipamente electrice - de ex. deschidere sau închidere întreruptor). Aplicaţia server interoghează fiecare IED conectat pentru a achiziţiona lista de evenimente generate în cadrul staţiei şi înregistrate de IED-uri în memoria internă.

Aceasta listă este comparată cu informaţiile existente deja în baza de date pentru a identifica noile evenimente.

Atunci când s-a identificat un eveniment nou este transmis id-ul (identificatorul) acestuia împreună cu id-ul IED-ului corespunzător către modulul de achiziţie evenimente. În cazul în care acest modul cade din diverse motive (eşuare achiziţie pachete de date pentru eveniment de la IED, eşuare salvare în baza de date, etc.), sunt prevăzuţi paşi anteriori unde aplicaţia EMCSIT Server se poate întoarce şi reîncepe procesul de achiziţie eveniment.

În cazul nostru, aplicaţia server va reinteroga IED-urile şi va compara informaţiile cu cele existente în baza de date şi apoi va relansa modulul de achiziţie evenimente. Dacă după câteva lansări succesive ale modulului de achiziţie evenimente în care nu s-a reuşit achiziţia, prelucrarea şi/sau salvarea unui anumit eveniment, acest lucru este semnalat utilizatorului în cadrul aplicaţiei şi se va trece la achiziţia următorului eveniment nou.

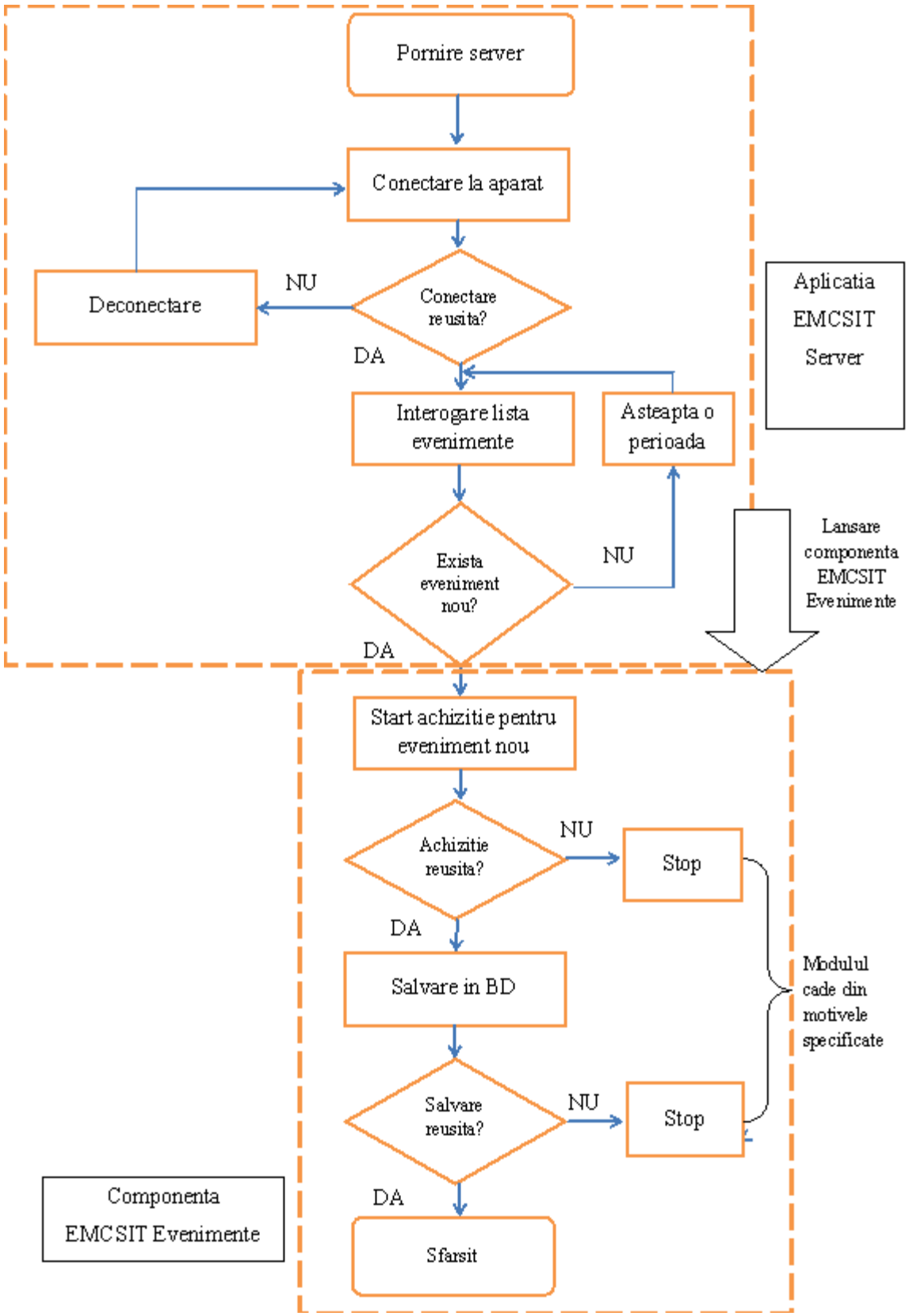

Fig. 4.5.1.1. Exemplu de implementare a blocurilor de recuperare în cadrul proiectului EMCSIT

4.5.2. Utilizarea tehnicilor de duplicare

Tehnicile de duplicare sunt utilizate în următoarele situaţii:

1) În cazul în care unul dintre echipamente nu mai funcţionează sau comunicaţia cu el eşuează, aplicaţia care comunică cu echipamentul trebuie să atenţioneze utilizatorul şi să încerce să obţină date echivalente de la alte echipamente.

În domeniul energetic, mai precis în cadrul unei staţii electrice de înaltă tensiune, echipamentele electrice primare sunt cele mai importante echipamente şi în special transformatorul/autotransformatorul de mare putere. Din cauza specificului lor, anumite informaţii achiziţionate prin intermediul echipamentelor de monitorizare, de la echipamentele primare, pot fi comune între mai multe tipuri de IED-uri. Spre exemplu, valorile obţinute pentru curenţii de linie se obţin atât prin IED-ul ce monitorizează întreruptoarele cât şi prin IED-ul ce monitorizează transformatoarele de măsură.

În acest caz, aplicaţia EMCSIT Staţie, ce prezintă informaţiile achiziţionate atât de la IED-ul corespunzător întreruptoarelor cât şi de la IED-ul corespunzător transformatoarelor de măsură, atunci când sesizează absenţa curenţilor de linie obţinuţi de la IED-ul pentru întreruptoare, va prezenta informaţiile similare achiziţionate de la IED-ul pentru transformatoarele de măsură.

Aceasta metodă nu previne toate cazurile de defecte sau căderi ce pot apărea dar cu siguranţă poate îmbunătăţi o arhitectură deja existentă care nu are implementată o astfel de metodă.

2) În cazul în care legătura la serverul central ce conţine baza de date nu este posibilă, aplicaţia EMCSIT Staţie va trebui să avertizeze utilizatorul şi în momentul în care legătura se restabileşte, să achiziţioneze ultimele informaţii înregistrate în baza de date centrală pentru a afişa informaţii actuale.

În cazul în care legătura dintre serverul central din camera de comandă unde se află baza de date centrală Oracle şi serverele locale se restabileşte, trebuie făcută sincronizarea cu bazele de date locale pentru a prelua informaţiile înregistrate pe toată perioada întreruperii legăturii.

În acest moment nu toate echipamentele dispun de memorie internă dar pentru acelea care au o astfel de posibilitate, în cazul în care a fost întreruptă comunicaţia cu aplicaţia de achiziţie şi prelucrare EMCSIT Server (I, S, D şi TC/TT), după remediere, aplicaţiile server vor trebui să descarce datele înregistrate din acea memorie internă şi să le salveze în baza de date locală.

Memoria internă a echipamentelor de monitorizare are posibilitatea de a stoca informaţiile achiziţionate de la echipamente electrice primare. În cazul în care legătura între serverele locale ce achiziţionează şi înregistrează datele de la echipamente de monitorizare nu este posibilă, în momentul reluării acesteia, aplicaţiile EMCSIT Server trebuie să achiziţioneze şi să afişeze informaţiile înregistrate în memoria internă a IED-urilor, privind parametrii monitorizaţi. În acest fel, informaţii actuale înregistrate se vor regăsi în bazele de date locale chiar dacă legătura cu aparatele de monitorizare nu a fost posibilă pentru o anumită perioadă de timp.

4.5.3. Utilizarea tehnicilor de reconfigurare şi reîntinerire (reconfiguration and rejuvenation)

Sistemul EMCSIT poate fi astfel configurat pentru ca fiecare server local sau serverul central să aibă posibilitatea la un moment dat, în funcţie de mai mulţi factori, să se reiniţializeze.

Reconfigurarea ţine seama de încărcarea procesorului, memoria internă liberă insuficientă, porturile de comunicaţie blocate, probleme de interfaţă de reţea, etc.

Soluţia o reprezintă alocarea unor resurse suplimentare pentru funcţionarea în continuare a sistemului. În cazul sistemului EMCSIT ce rulează pe o platforma Microsoft Windows, o soluţie extremă este programarea la un moment dat a unei reporniri pentru serverul respectiv pe care rulează aplicaţia.

Reîntinerirea reprezintă o tehnică ce ia în considerare faptul că aplicaţia EMCSIT SuperServer sau Staţie poate intra într-o buclă infinită sau să atingă un maxim al resurselor utilizate datorită unor diverse cauze. Este prevăzut ca în cazul unei reporniri accidentale (lipsa alimentarii cu energie electrică a serverelor pentru o perioada ce depăşeşte autonomia UPS-urilor conectate) sau voite, aplicaţia EMCSIT SuperServer sau Staţie să se execute automat în momentul încărcării sistemului de operare pe server sau calculatorul client. Deşi sistemul rulează într-o staţie electrică şi teoretic întreruperea alimentarii cu energie electrică este practic imposibilă, s-a decis totuşi ca această posibilitate să fie implementată.

Totodată, dacă se deschid două sau mai multe instanţe ale aplicaţiei, trebuie făcută o verificare şi o închidere a proceselor respective deoarece acest lucru conduce la blocarea sistemului.

4.6. Concluzii

În acest capitol au fost prezentate metode pentru asigurarea calităţii software a unui sistem folosit ca studiu de caz: sistemul de monitorizare şi control EMCSIT (Echipament pentru Monitorizarea Complexă a Staţiilor de Înaltă Tensiune).

A fost prezentată arhitectura sistemului şi apoi problemele care au fost identificate în dezvoltarea şi utilizarea acestui sistem.

Pornind de la analiza consecinţelor testării insuficiente a sistemului înainte de instalarea sa în mediul real de funcţionare (cu echipamentele electrice cuplate la echipamentele de monitorizare), am dezvoltat un simulator de IED-uri care permite o testare acoperitoare a aplicaţiilor server din cadrul sistemului EMCSIT.

Pentru eficientizarea procesului de testare, în cadrul simulatorului s-a implementat metoda *Pairwise Testing* de generare a datelor de test. Metoda permite depistarea a până la 70% din defectele existente într-o aplicaţie, utilizând mult mai puţine cazuri de test faţă de metoda de testare care acoperă toate combinaţiile posibile ale valorilor selectate pentru parametrii monitorizaţi.

Sunt menţionate şi tehnici pentru asigurarea securităţii sistemului EMCSIT, axate în special pe protejarea accesului la aplicaţiile software.

În cadrul sistemului EMCSIT au fost implementate tehnici de toleranţă la defecte cum ar fi:

- folosirea de blocuri de recuperare;
- utilizarea tehnicilor de duplicare;
- utilizarea tehnicilor de reconfigurare şi reîntinerire.

5. Standardul IEC61850

5.1. Introducere

Staţiile de transformare reprezintă componente cheie ale reţelei electrice, facilitând transportul şi distribuţia energiei electrice. Au un rol vital în monitorizarea şi controlul fluxului de energie electrică şi asigură interconectarea între companiile de electricitate, reţelele de transport şi distribuţie şi consumatorii finali.

Sistemele de automatizare ale staţiilor electrice fac posibilă monitorizarea şi controlul în timp real şi ajută la maximizarea eficienței, siguranţei şi a integrităţii datelor.

Ultimii ani au adus dezvoltări semnificative în ceea ce priveşte standardele ce definesc comunicaţia la nivel de staţii electrice – probabil cel mai important pas în acest sens a fost publicarea standardului IEC 61850. Tot mai mulţi producători şi-au adaptat deja produsele pentru a fi conforme cu noul standard sau chiar au dezvoltat produse noi pornind de la modelul de date pe care-l defineşte standardul. Noul standard este tot mai mult impus şi de către diverşi beneficiari. Viitorul în comunicaţii la nivel de staţii electrice este deja prefigurat de IEC 61850.

Cerinţele de performanţă într-o reţea de comunicaţii din staţie, depind de dimensiunea staţiei şi importanţa ei în sistemul energetic. Există următoarele tipuri de clasificări conform acestui standard:

- Staţii mici de distribuţie;
- Staţii medii de distribuţie;
- Staţii mari de distribuţie;
- Staţii mici de transport;
- Staţii mari de transport;
- Variante combinate;

Standardul IEC61850 conţine un set de documente care se axează pe următoarele aspecte majore: un model funcţional al domeniului de aplicare pentru SA (Substation Automation - automatizările din staţia electrică) – partea a 5-a, un model de date pentru SAS (Substation Automation System - sistemul de automatizări din staţie), protocoalele de comunicaţii şi serviciile aferente – partea a 7-a şi părţile 8 şi 9, un limbaj de configurare a staţiei bazat pe XML (SCL – Substation Configuration Language) – partea a 6-a.

Cerinţele pieţii, îndeplinite de standardul IEC61850, sunt:

- interoperabilitatea, astfel că standardul trebuie să suporte toate funcţiile în aplicaţiile staţiei. Interoperabilitatea presupune ca diferite IED-uri de la diverşi producători să poată fi capabile să schimbe şi să utilizeze informaţii în timp real fără convertoare de protocol.
- arhitectura deschisă, ce poate include operaţiuni implementate de diversele companii de electricitate din toată lumea. Permite o alocare arbitrară a funcţiilor către dispozitive şi acceptă sisteme centralizate şi descentralizate.
- stabilitate pe termen lung, referindu-se la durata de viaţă a unei staţii (a echipamentelor primare) care este cuprinsă între 40 şi 60 de ani, iar echipamentele din automatizarea staţiei vor fi schimbate de două-trei ori în această perioadă. În timp, staţia trebuie să integreze noile componente de la acelaşi producător sau de la alţi producători.

IED-ul reprezintă, conform IEC61850 orice echipament ce include unul sau mai multe procesoare (microcontrollere) cu posibilitatea de a primi sau trimite date/control de la sau către o sursă externă.

Acest standard de comunicaţii promite să revoluţioneze automatizările din staţiile electrice cu mesaje peer-to-peer foarte rapide, date structurate şi orientate-obiect.

Este destinat să asigure un singur protocol pentru o întreagă staţie, să implementeze un format obişnuit pentru descrierea staţiei, să faciliteze modelarea datelor necesare staţiei, să definească serviciile de bază necesare pentru a transfera datele prin diferite protocoale de comunicaţie şi să permită interoperabilitatea între echipamente ale diverşilor producători.

O dată cu evoluţia standardului, s-au definit două servicii critice în timp: transmisia rapidă a semnalelor de tipul căderilor şi eşantionarea valorilor de curent şi tensiune analogice. Acestea au dus la extensia legăturilor seriale între dispozitive electronice inteligente (IED) şi interfeţe electronice, considerând timpul de transfer redus, la 3 microsecunde şi timpul de sincronizare de 1 microsecundă.

Modelul de date bazat pe obiecte este independent de aplicaţia în care se utilizează. Clasele de obiecte sunt legate de domeniul staţiei electrice iar obiecte model pentru energia eoliană, energia hidro, sursele de energie distribuită au fost adăugate ulterior în standard.

Toate funcţiile modelului de date, incluzând structurarea datelor pentru echipamentele primare ale staţiei sunt împărţite pe module care apoi pot fi implementate separat în IED-uri.

Elementele de bază se numesc noduri logice (LN). Fiecare obiect va cuprinde atribute, ce pot reprezenta valori sau caracteristici detaliate ale echipamentelor.

Operaţiile suportate de către modelul de date definit de standardul IEC61850 sunt următoarele:

- citirea datelor ca valori ale atributelor;
- scrierea unei valori drept atribut al configurării;
- controlul dispozitivelor de comutaţie şi a altor obiecte prin metode precum „selectare înainte de operare” sau „operare directă”;
- raportarea unui eveniment care duce la modificarea unor valori;
- stocarea evenimentelor importante şi a altor date de istoric;
- transfer de fişiere pentru configurare şi istoric;
- GOOSE (Generic Object Oriented Substation Event) reprezintă acronimul pentru un eveniment de sistem generic orientat pe obiecte şi este un serviciu pentru transmisii de viteză a informaţiilor critice precum modificări de stare a echipamentului monitorizat, blocaje între IED, etc.;
- eşantionarea valorilor prin transmiterea unui şir de valori sincronizate de tensiune şi curent.

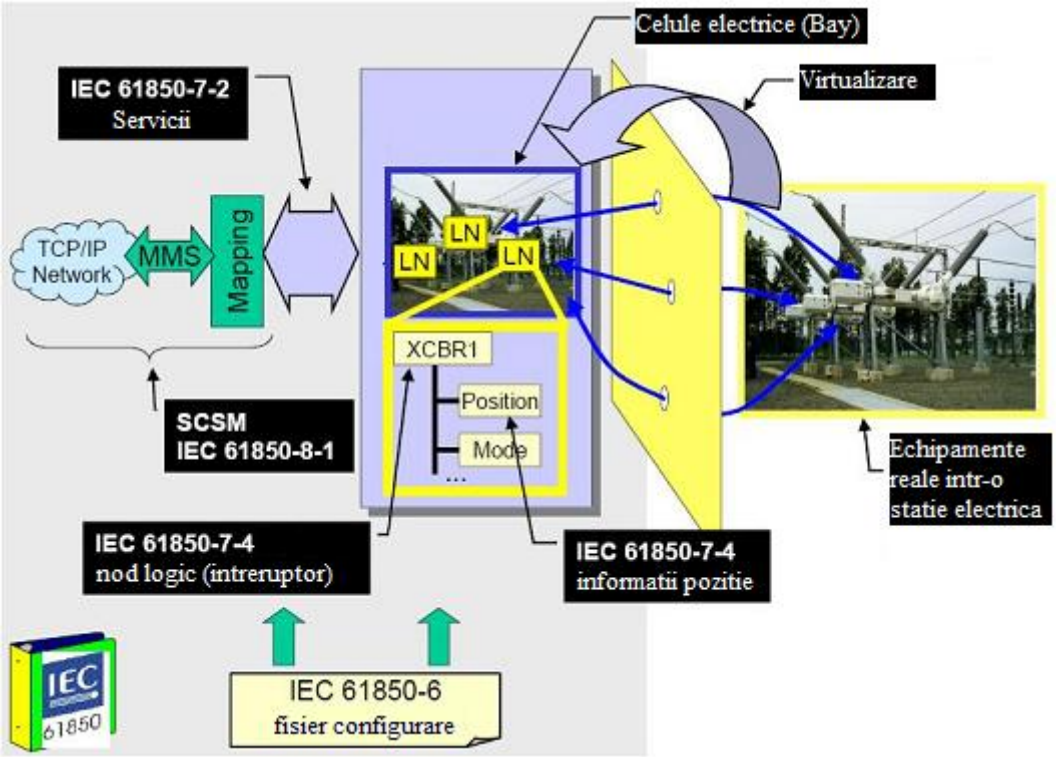

Fig. 5.1.1. Modelarea conceptuală conform standardului IEC61850 [Iec9-10]

Standardul IEC61850 utilizează Ethernet ca tehnologie de comunicaţie de bază, la o viteză de 100Mb/s sau mai mare în funcţie de elementele de reţea disponibile. Se utilizează topologii de reţea tip arbore şi inel pentru switch-urile de reţea. Cea mai utilizată topologie de reţea este cea de tip inel cu reconfigurarea automată în cazul unei pierderi de legătură sau a unui defect al switch-urilor. Magistrala staţiei conectează IED-urile pentru control, protecţie şi monitorizare cu dispozitivele de la nivelul unităţilor terminale.

Limbajul de configurare a staţiei (SCL) reprezintă una dintre cele mai importante realizări ale standardului IEC61850. SCL face posibilă crearea de fişiere utilizate pentru schimbul de date de configurare între aplicaţii software specializate. Descrierea configuraţiei sistemului reprezintă un tip de fişier definit de standardul IEC61850 şi constituie cel mai important document al unui sistem de automatizare complet al staţiei. Un aspect foarte important al integrării standardului IEC61850 constă în alegerea dispozitivelor electronice ce sunt conforme cu standardul.

Viitoarele reţele electrice cu generarea descentralizată a energiei, cu o putere de cumpărare flexibilă, cu o încredere mai mare în reţea sunt catalogate drept reţele inteligente (Smart Grid). Mai multe informaţii vor fi disponibile într-o manieră mai sigură şi mai scurtă ca timp, către mai multe aplicaţii şi utilizatori distribuiţi, iar controlul va fi optimizat. Datele vor fi disponibile într-o reţea informală de date, în conformitate cu semantica standardizată a datelor.

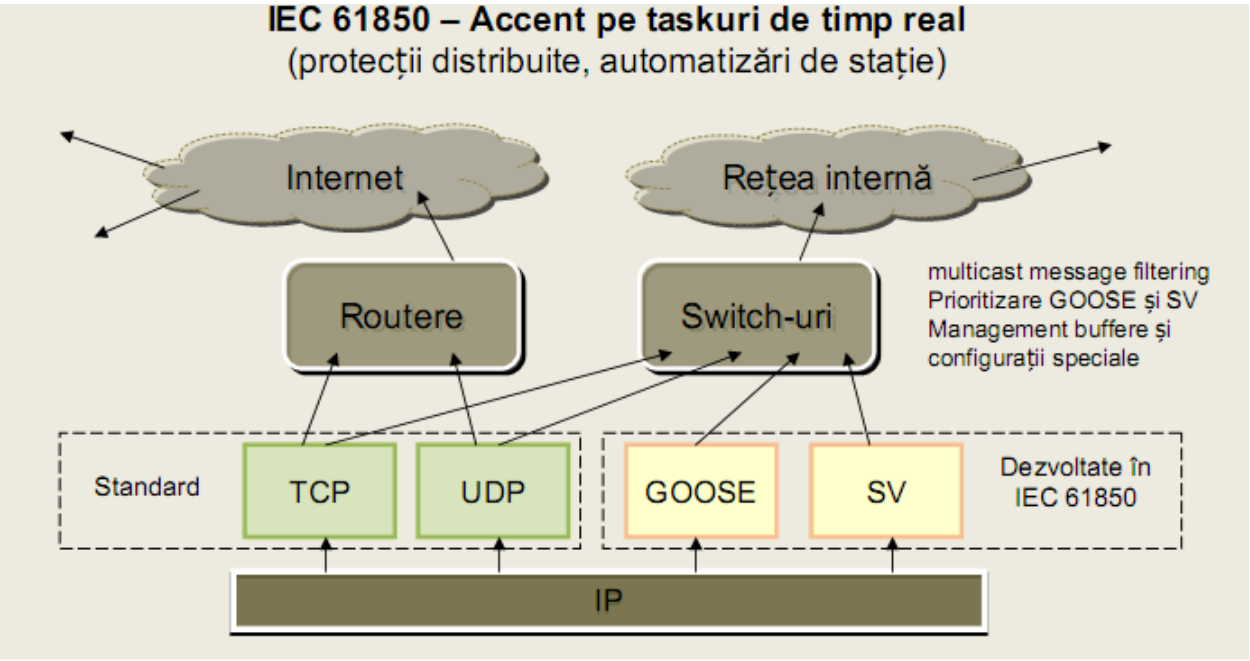

Fig. 5.1.2. Standarde pentru comunicarea în interiorul unei staţii electrice [Iec9-1]

Pentru comunicarea în cadrul reţelei LAN, se utilizează protocoalele dedicate IEC61850: GOOSE şi SV iar pentru comunicarea în cadrul Internet se folosesc protocoalele standard TCP şi respectiv UDP.

5.2. Limbajul SCL

Substation Configuration Language (SCL – limbajul de configurare a staţiei) este limbajul şi formatul de reprezentare specificat de IEC61850 pentru descrierea configuraţiei echipamentelor dintr-o staţie electrică. Acesta include reprezentarea datelor modelate şi a serviciilor de comunicaţie specificate de documentele standardului IEC61850-7-X. Descrierea completă a SCL şi detaliile sunt specificate în documentul standard IEC 61850-6 [Iec9-3]. Include reprezentarea datelor pentru echipamentele din staţie, funcţiile asociate fiecărui echipament - reprezentate ca noduri logice, sisteme şi capabilităţi de comunicaţie. Reprezentarea completă a datelor în limbajul SCL oferă posibilitatea interoperabilităţii echipamentelor dintr-o staţie electrică prin schimb de fişiere.

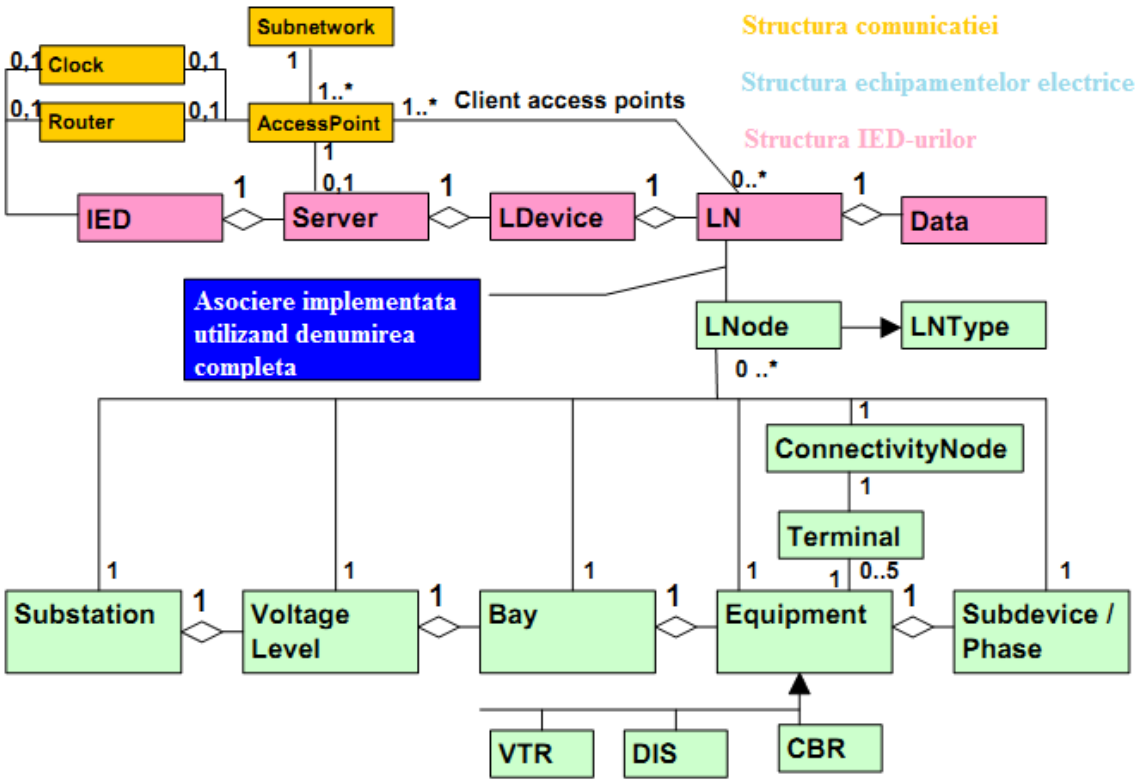

Fig. 5.2.1. Proiectarea unei staţii electrice cu ajutorul SCL, conform [Iec9-9]

Marele avantaj al utilizării IEC61850 este interoperabilitatea între IED-uri aparţinând diverşilor producători din domeniu. Limbajul are o convenţie de denumire standard a datelor, echipamentele sunt descrise automat prin intermediul limbajului SCL, permite modelarea virtuală a echipamentelor logice şi oferă un limbaj comun de configurare a echipamentelor.

În anexa 1 se prezintă structura generală a unui fişier SCL şi paşii modelării unei staţii electrice folosind acest limbaj.

5.3. Concluzii

În cadrul acestui capitol am descris standardul IEC61850 şi limbajul SCL de configurare a unei staţii electrice.

Acest standard a fost ales întrucât este unanim acceptat de producătorii de echipamente de monitorizare din domeniu. Sunt prezentate aspecte generale privind standardul IEC61850 şi utilizarea acestuia în cadrul unei staţii electrice.

IED-ul reprezintă, conform IEC61850, orice echipament ce include unul sau mai multe procesoare (microcontrollere) cu posibilitatea de a primi sau trimite date/control de la sau către o sursă externă.

Acest standard de comunicaţii promite să revoluţioneze automatizările din staţiile electrice cu mesaje peer-to-peer foarte rapide, date structurate şi orientate-obiect.

Limbajul SCL permite descrierea echipamentelor electrice din cadrul unei staţii precum şi asocierea IED-urilor destinate monitorizării şi respectiv controlului acestora.

Limbajul SCL a fost ales în continuare pentru configurarea unei staţii electrice din studiul de caz, ce include modulele unui sistem de monitorizare şi control (anexa 2).

6. Utilizarea unor modele matematice pentru estimarea fiabilităţii sistemelor de monitorizare şi control a staţiilor electrice

6.1. Modelul de distribuţie Rayleigh

Modelul Rayleigh este un model parametric, care se bazează pe o distribuţie statistică specifică. Atunci când parametrii distribuţiei statistice sunt estimaţi pe baza datelor dintr-un proiect software, pot fi făcute previziuni referitoare la rata de defectare a produsului software.

Modelul Rayleigh este unul dintre cele mai folosite modele matematice pentru modelarea şi predicţia ratelor de defectare a produselor software în timp ([Urs8-1], [Urs8-2]).

Modelul Rayleigh este un caz particular al modelului de distribuţie Weibull. Acesta este folosit de foarte mulţi ani în diverse domenii de inginerie pentru analiza fiabilităţii. Una dintre caracteristicile sale este faptul că partea finală a funcţiei sale de densitate de probabilitate se apropie asimptotic de zero, dar niciodată nu atinge aceasta limită. Funcţia sa de distribuţie cumulativă (CDF) şi funcţia de densitate a probabilităţii (PDF) au următoarele expresii:

$$F(x) = 1 - e^{-x^2/2\sigma^2}$$

şi respectiv

$$f(x) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2}, x \geq 0$$

unde σ este parametrul de scală al modelului Rayleigh şi x este variabila aleatoare.

În funcţie de valorile parametrului σ, se modifică graficul funcţiei de distribuţie. Acesta reprezintă variaţia formulei ce exprimă funcţia de densitate dacă se modifică unitatea de măsură pe abscisă.

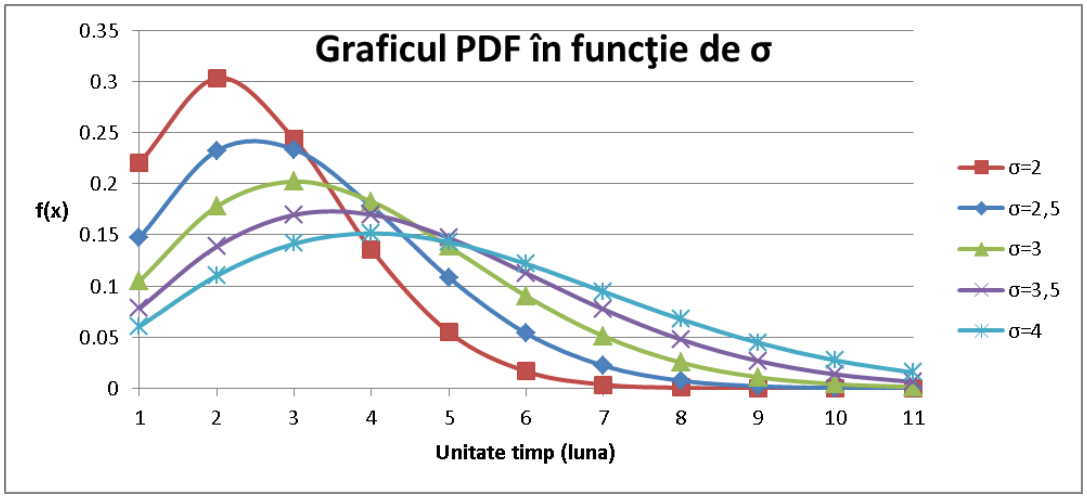

Fig. 6.1. Graficul pentru funcţia de densitate a probabilităţii în funcţie de valoarea parametrului σ

Atunci când se aplică pentru produse software, PDF (funcţia de densitate a probabilităţii) se referă la densitatea defectelor (rata de defectare) în timp iar CDF reprezintă funcţia de repartiţie sau distribuţia cumulativă de probabilitate [Ban9-12].

Există o bază de date empirice foarte mare privind aplicarea modelului Rayleigh. S-a demonstrat că ciclul de viaţă al proiectelor software urmează modelul descris de curba Rayleigh, din punctul de vedere al defectelor descoperite şi al remedierii acestora.

Modelul Rayleigh este un model parametric formal, care poate fi utilizat pentru estimarea defectelor ramase intr-un produs software atunci când activitatea de dezvoltare s-a încheiat, iar produsul este gata pentru a fi livrat clienţilor. Interpretarea rezultatelor obţinute cu ajutorul modelului Rayleigh contribuie la dezvoltarea eficientă de software.

Am aplicat modelul Rayleigh pe datele înregistrate în timpul dezvoltării aplicaţiilor server pentru cele 5 tipuri de IED-uri prezentate în studiul de caz EMCSIT (cap. 4).

Graficele reprezentând numărul de defecte constatate în timpul dezvoltării aplicaţiilor server pentru diferite tipuri de IED-uri sunt următoarele:

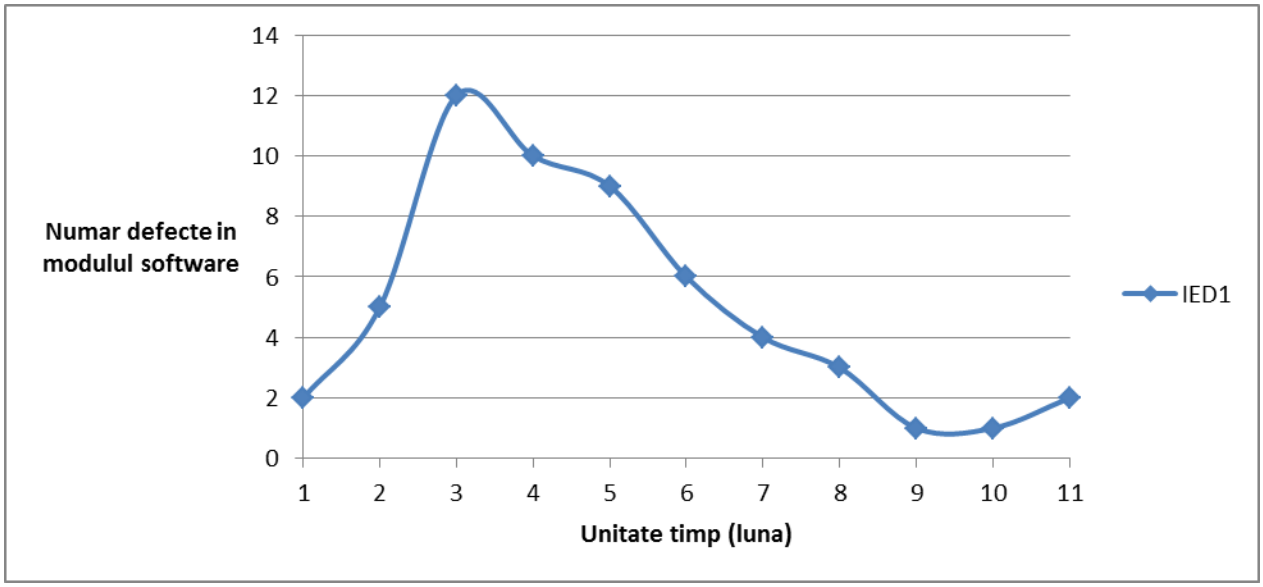

Fig. 6.1.1. Reprezentarea grafică a numărului de defecte pentru aplicația tip server pentru IED-urile ce monitorizează întreruptoare (IED1)

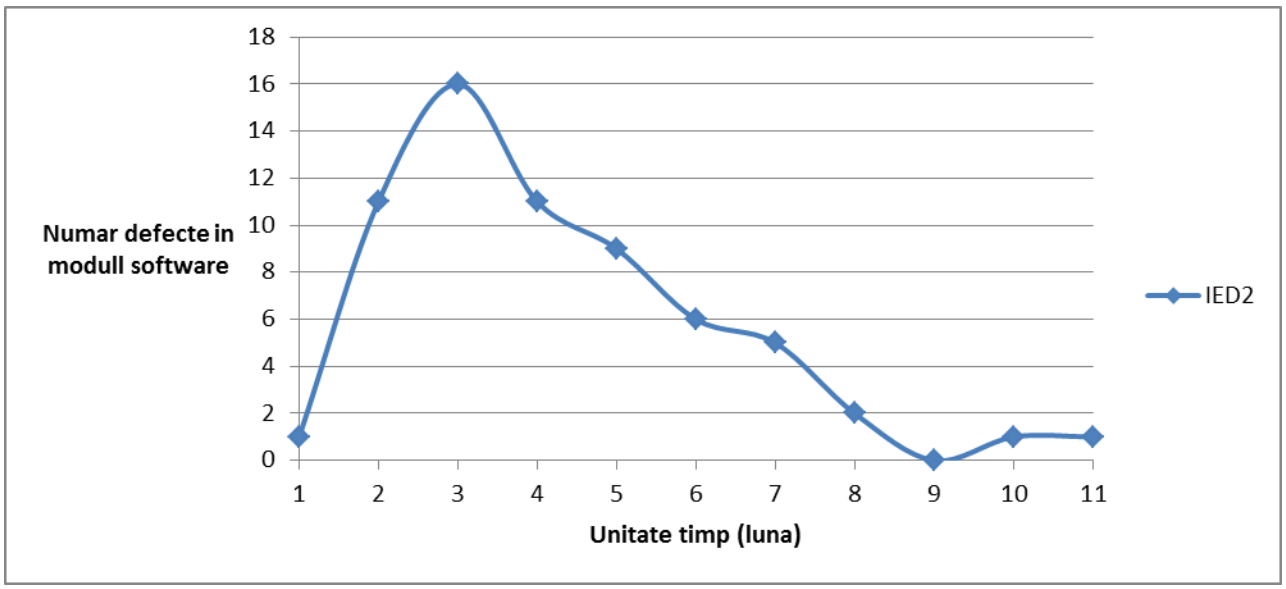

Fig. 6.1.2. Reprezentarea grafică a numărului de defecte pentru aplicația tip server pentru IED-urile ce monitorizează separatoare (IED2)

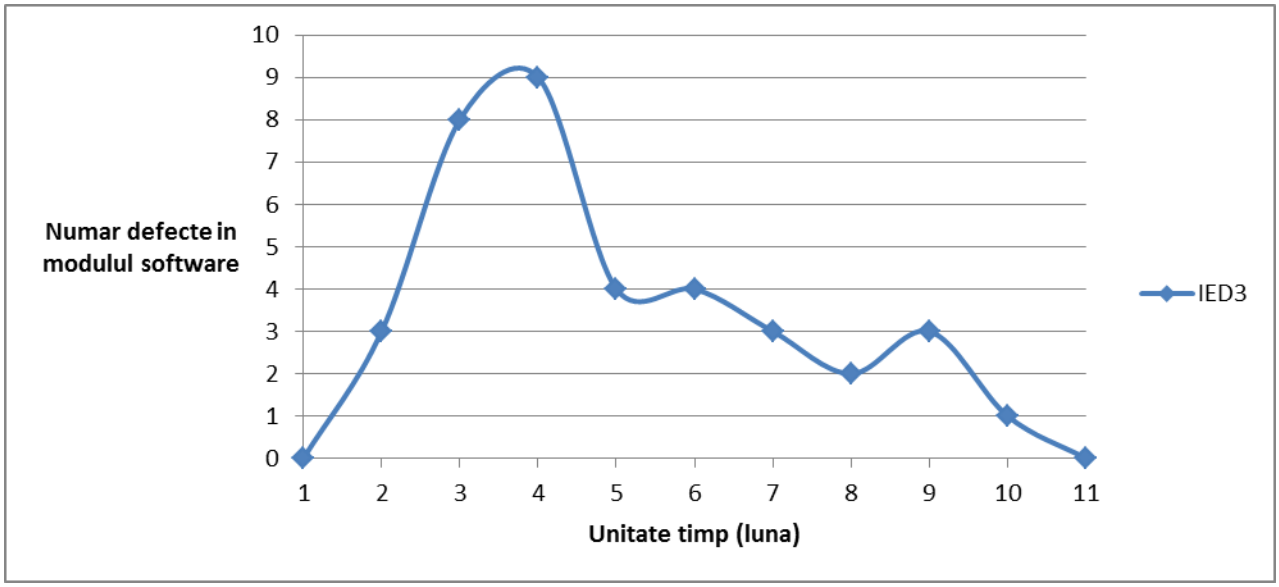

Fig. 6.1.3. Reprezentarea grafică a numărului de defecte pentru aplicația tip server pentru IED-urile ce monitorizează descărcătoare (IED3)

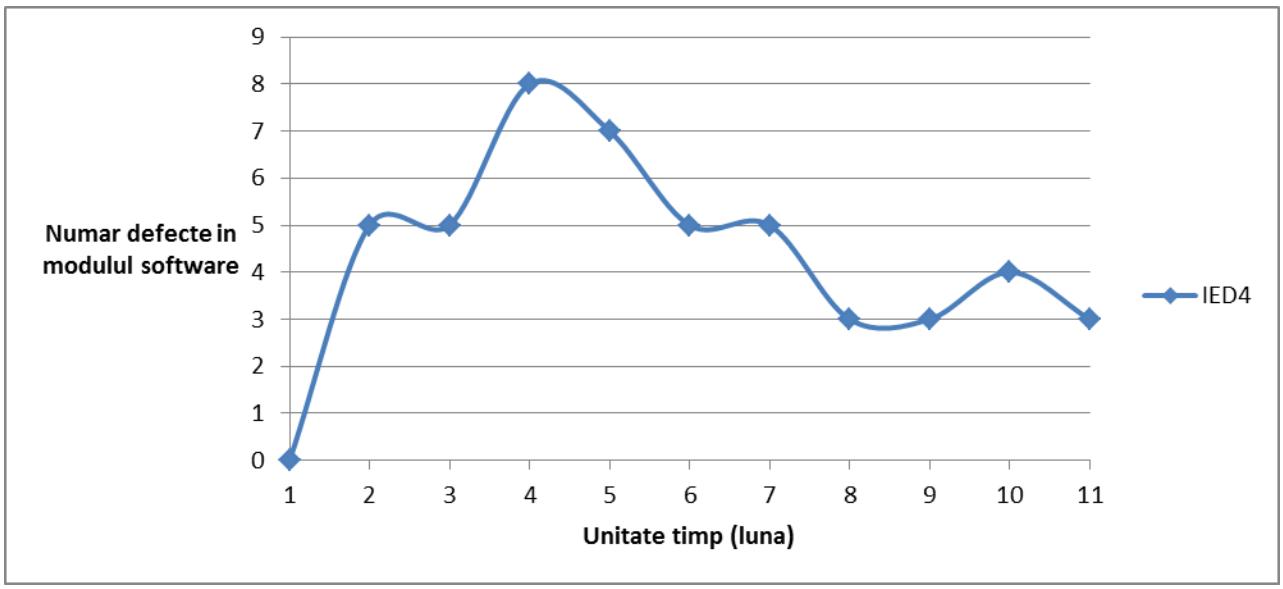

Fig. 6.1.4. Reprezentarea grafică a numărului de defecte pentru aplicația tip server pentru IED-urile ce monitorizează transformatoare de măsură (IED4)

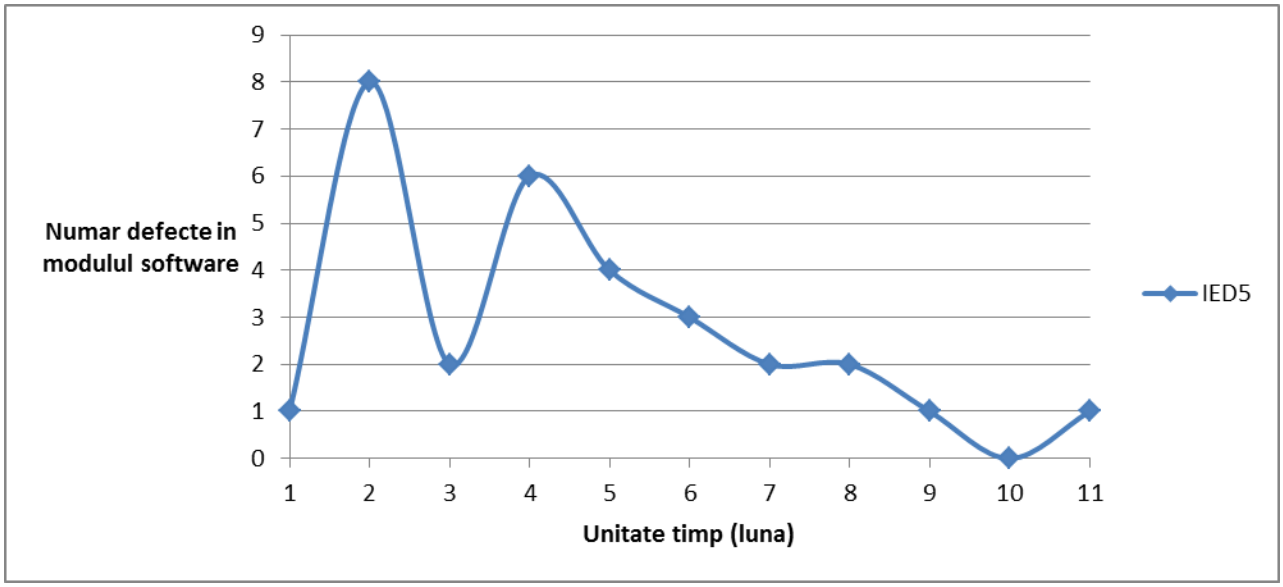

Fig. 6.1.5. Reprezentarea grafică a numărului de defecte pentru aplicaţia tip server pentru IED-urile ce monitorizează transformatoare de putere/bobine de compensare (IED5)

Primele 2-3 luni au reprezentat perioada de design şi implementare a aplicaţiilor server pentru fiecare tip de IED. În aceasta perioadă au apărut primele defecte, care au fost remediate.

Următoarele 4-5 luni au fost utilizate pentru testarea aplicaţiilor server utilizând IED-urile dezvoltate dar fără a fi conectate la echipamentele electrice. Aceasta testare a avut loc înainte de instalarea IED-urilor în staţia electrică. Pentru anumite tipuri de aplicaţii server s-a demonstrat că această perioadă de testare a fost insuficientă.

În ultimele 4 luni, IED-urile au fost montate în staţie şi conectate la echipamentele electrice monitorizate. Au fost instalate aplicaţiile server (pentru tipurile de IED1, 2, 3 şi respectiv 4) şi aplicaţia EMCSIT SuperServer ce utilizează cele 4 tipuri de aplicaţii server pe calculatoarele server locale, au fost configurate cele două servere centrale de baze de date aferente celor două substaţii şi a fost instalată şi configurată aplicaţia EMCSIT Staţie pe calculatorul client din camera de comandă.

Estimarea ratei defectelor în funcţionarea aplicaţiilor de tip server din sistemul EMCSIT

Datele primare reprezintă numărul de defecte înregistrate în fiecare lună pentru aplicaţiile tip server pentru IED1, IED2, IED3, IED4 şi respectiv IED5. Pe abscisă este

precizată luna în care au fost colectate informaţiile privind numărul de defecte iar pe ordonată este reprezentată valoarea funcţiei de densitate a probabilităţii.

S-au obţinut următoarele rezultate:

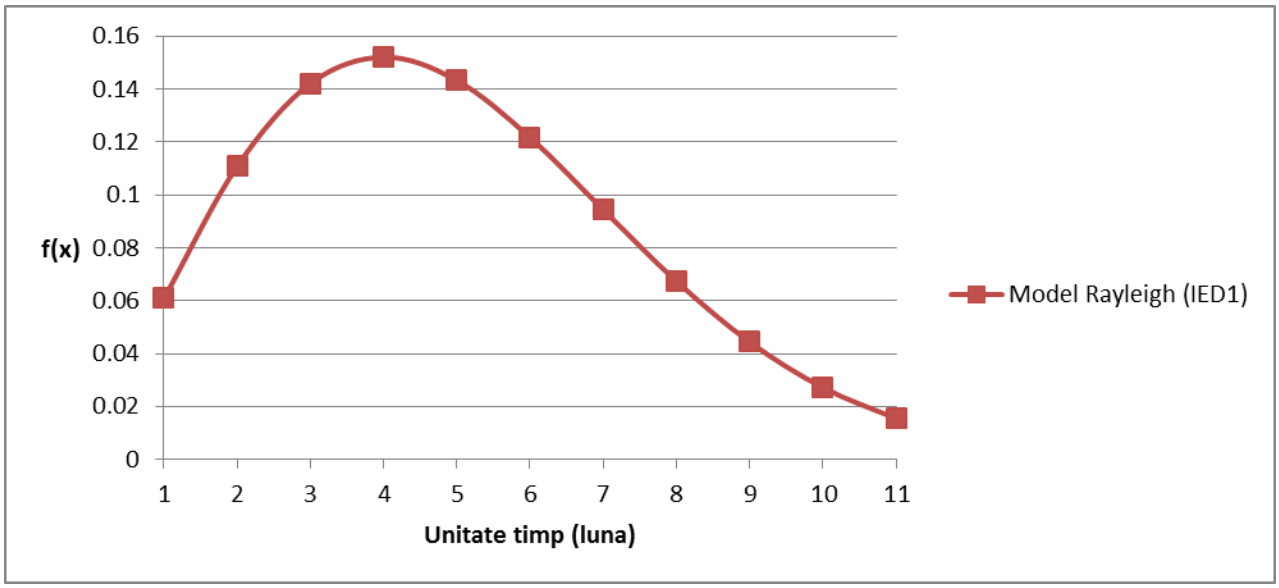

Fig. 6.1.6. Aplicarea modelului de distribuţie Rayleigh pentru aplicaţia tip server IED1

Pentru aplicaţia tip server IED1, în perioada de proiectare şi implementare au fost descoperite puţine defecte, în timp ce majoritatea defectelor au fost descoperite la începutul perioadei de testare cu serverul conectat la IED, acesta nefiind conectat la echipamentul electric monitorizat. După instalarea în staţie, în perioada de testare cu IED-urile conectate la echipamentele electrice au fost identificate puţine defecte. Spre sfârşitul perioadei de testare în staţie, numărul defectelor descoperite a scăzut şi aproape s-a stabilizat ajungând la valori mici.

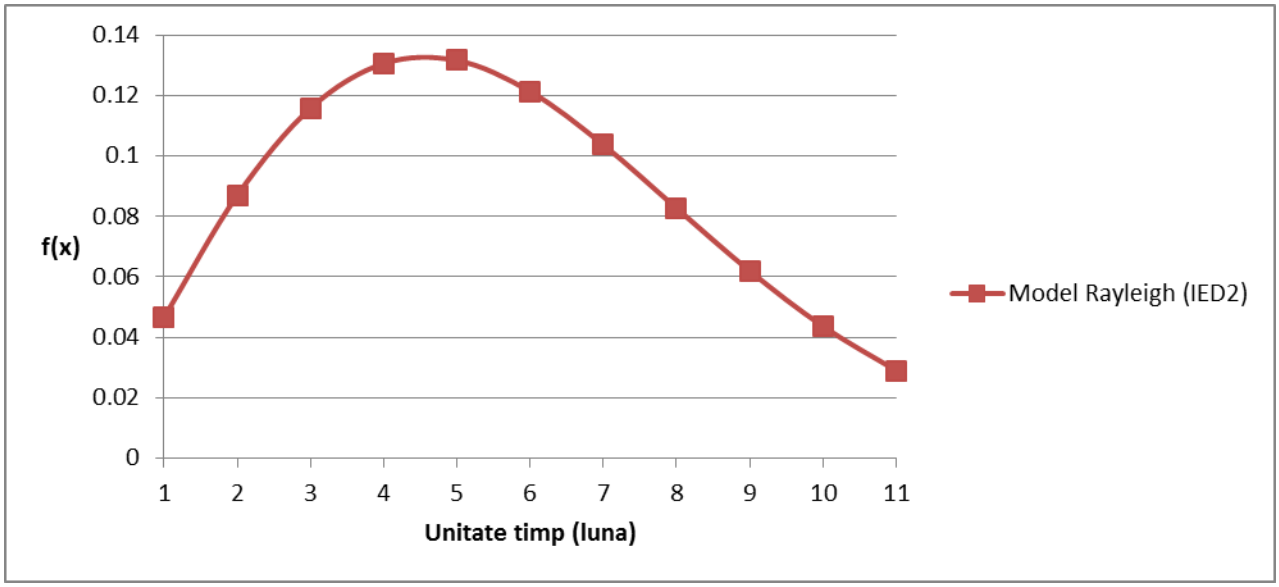

Fig. 6.1.7. Aplicarea modelului de distribuţie Rayleigh pentru aplicaţia tip server IED2

Pentru aplicaţia tip server IED2, majoritatea defectelor au fost descoperite la sfârşitul perioadei de proiectare şi implementare şi la începutul perioadei de testare cu conectare la IED neconectat la echipamentul electric monitorizat. Spre sfârşitul perioadei de testare în staţie, numărul defectelor descoperite a scăzut, dar nu s-a stabilizat.

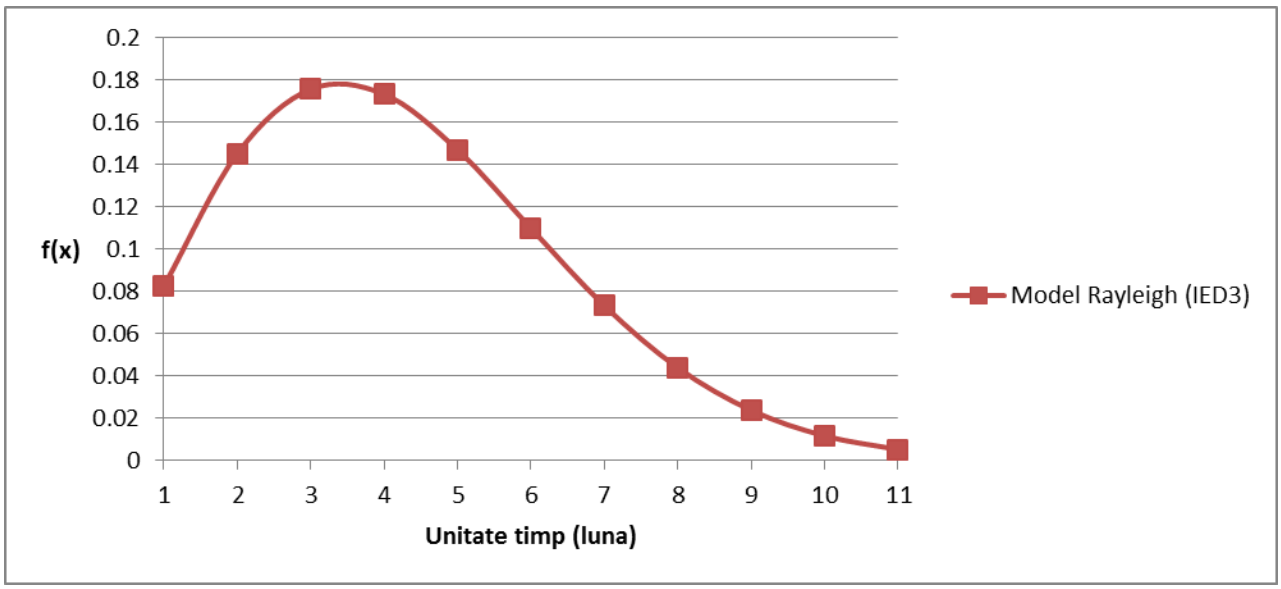

Fig. 6.1.8. Aplicarea modelului de distribuţie Rayleigh pentru aplicaţia tip server IED3

Pentru aplicaţia tip server IED3, spre sfârşitul perioadei de proiectare şi implementare (lunile 2-3) au fost descoperite majoritatea defectelor, comparativ cu restul perioadelor. În perioada de testare cu conectare la IED dar neconectate la echipamentul electric monitorizat

(lunile 4-7) şi în perioada de testare cu IED-urile conectate la echipamentul electric (lunile 8-11) numărul defectelor descoperite a scăzut, atingând valori mici şi aproape stabilizându-se către sfârşit.

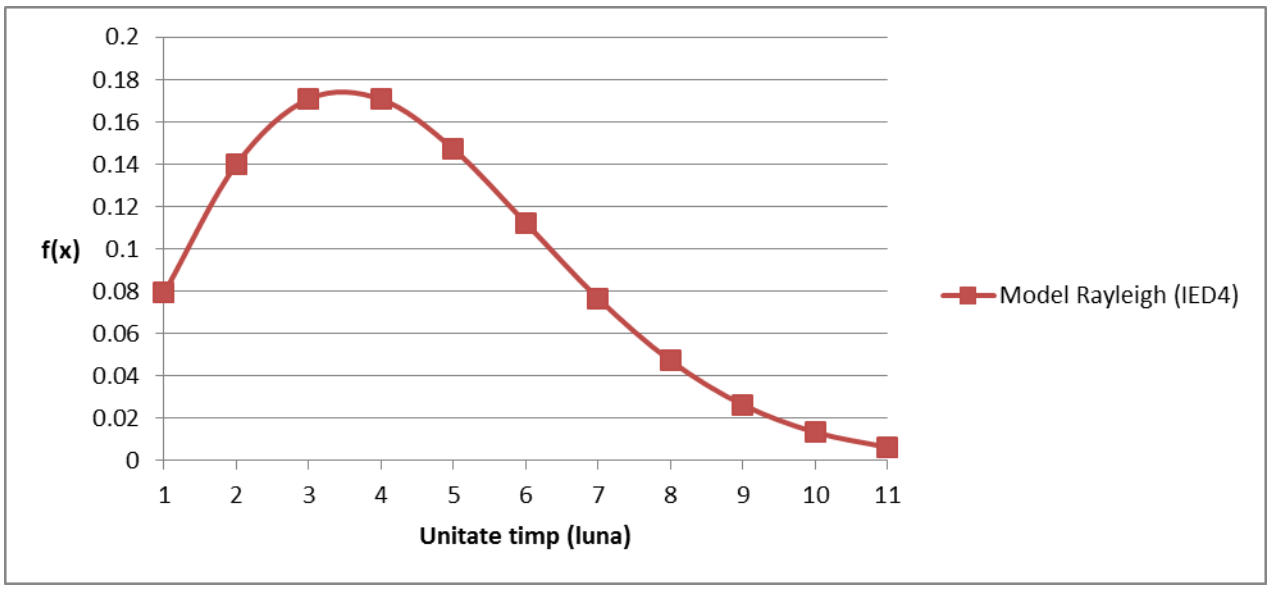

Fig. 6.1.9. Aplicarea modelului de distribuţie Rayleigh pentru aplicaţia tip server IED4

Pentru aplicaţia tip server IED4, în perioada de proiectare şi implementare (lunile 2-3) au fost descoperite şi remediate multe defecte. Totodată a mai fost descoperit un număr semnificativ de defecte la începutul perioadei de testare folosind IED-ul dar neconectat la echipamentul electric monitorizat (lunile 4-5). Similar cu cazul aplicaţiei tip server IED3, în perioada de testare în staţie cu IED-urile conectate la echipamentele electrice (lunile 8-11), numărul defectelor descoperite a scăzut, atingând valori mici şi aproape stabilizându-se către sfârşitul perioadei.

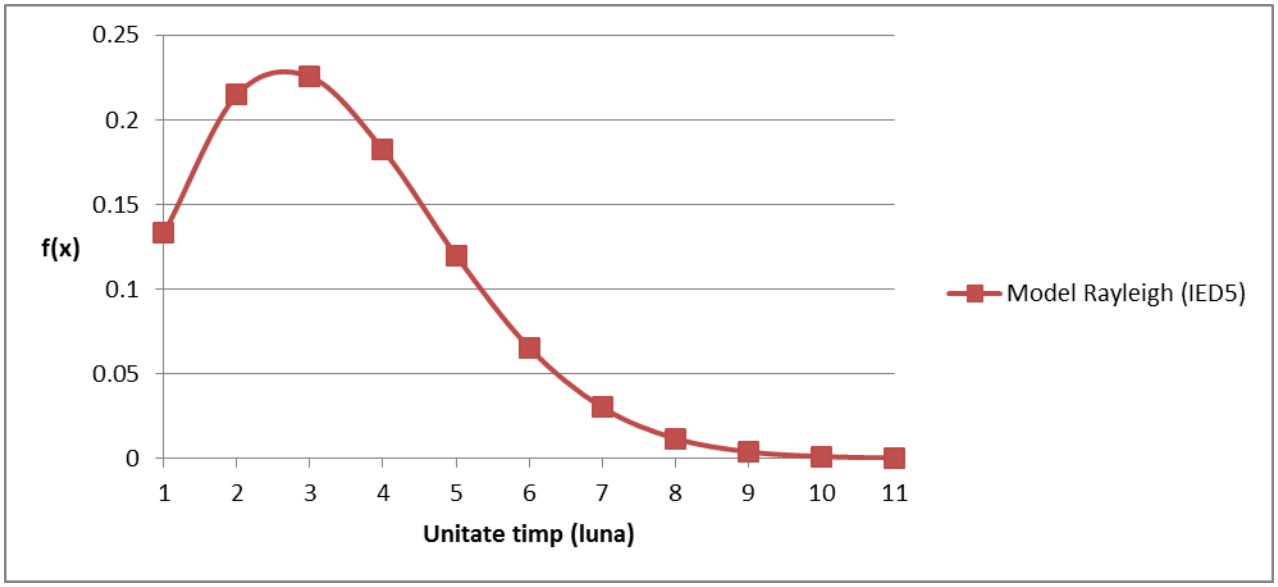

Fig. 6.1.10. Aplicarea modelului de distribuţie Rayleigh pentru aplicaţia tip server IED5

Pentru aplicaţia tip server IED5, în perioada de proiectare şi implementare (primele 4 luni) au fost descoperite majoritatea defectelor. Numărul acestora a scăzut în perioada de testare cu conectare la IED dar neconectate la echipamentul electric monitorizat. Spre sfârşitul perioadei de testare în staţie cu IED-urile conectate la echipamentul electric numărul defectelor descoperite s-a stabilizat, ajungând la valori foarte mici (lunile 10-11).

În urma aplicării modelului Rayleigh pentru aplicaţiile server asociate celor 5 tipuri de IED-uri, s-a constatat că pentru serverele asociate dispozitivelor IED3, IED4 şi IED5 rata descoperirii defectelor pe parcursul dezvoltării a fost bună, astfel încât la sfârşitul perioadei de testare rata defectelor s-a stabilizat şi a avut o valoare mică.

Modelarea cu ajutorul curbei Rayleigh a ratei defectelor pentru serverul IED1 indică faptul că la sfârşitul perioadei de testare rata defectelor a scăzut dar nu s-a stabilizat încă, fiind posibilă descoperirea unor noi defecte în perioada imediat următoare.

Pentru serverul IED2, la sfârşitul perioadei de testare rata defectelor a fost mare, numărul de defecte rămase în software a fost de asemenea mare, indicând faptul că testele efectuate au fost insuficiente.

Concluzia generală în urma aplicării modelului Rayleigh este că în situaţia în care s-ar fi utilizat simulatorul software de la începutul proiectului, se puteau descoperi mai multe defecte înainte de instalarea în staţie.

6.2. Modelul matematic al lanţului Markov

În teoria probabilităţilor, un model Markov este un proces stochastic ce posedă proprietatea Markov. Această proprietate se referă la faptul că stările viitoare ale unui sistem modelat printr-un lanţ Markov depind de cele prezente şi sunt independente de cele trecute.

Lanţul Markov modelează starea unui sistem cu o variabilă aleatoare care se schimbă în timp. Astfel, la fiecare moment, sistemul îşi poate schimba sau păstra starea, în conformitate cu o anumită distribuţie de probabilitate. Schimbările de stare sunt numite tranziţii.

Lanţul Markov omogen simplu arată că probabilitatea ca una din variabilele aleatoare x_n să ia o valoare a_i depinde numai de variabila precedentă (valoarea ei) şi nu depinde de toate valorile anterioare. Se mai spune că variabila x_n depinde de cel mai „apropiat trecut" x_{n-1}. În cazul unui lanţ Markov multiplu, probabilitatea ca variabila x_n să ia o valoare a_i depinde numai de valorile lanţului de m variabile precedente, fiind independentă de valorile luate de restul variabilelor. Dacă variabila x_n a lanţului ia valoarea a_i, atunci variabilele următoare x_{n+1}, x_{n+2}, ... depind numai de valoarea a_i luată de variabila x_n.

De aceea, dezvoltarea viitoare a lanţului depinde numai de prezent, fiind independentă de trecut.

Un lanţ Markov aplicat în ingineria software presupune existenţa a doua stări pentru aplicaţia software: starea bună şi starea de cădere.

Pe baza experienţei personale de dezvoltare şi utilizare a unor sisteme software de monitorizare şi control, propun ca un astfel de sistem să fie asociat la un moment dat cu una dintre următoarele stări:

1. Stare bună:
   - o sistemul în ansamblu funcţionează conform specificaţiilor;
2. Stare acceptabilă:
   - o există probleme la achiziţionarea pachetelor de date. Sunt pachete ratate la anumite intervale de timp;
   - o rarele erori nu presupun încă intervenţia personalului din staţie;
3. Stare proastă:
   - o sistem indisponibil majoritatea timpului;
   - o există pachete de date ratate (neachiziţionate de către aplicaţiile server) foarte des;
   - o întârzieri în actualizarea datelor afişate;

- este necesară intervenţia utilizatorului pentru repornirea uneia sau mai multor aplicaţii din sistem;
- poate fi necesară chiar repornirea calculatorului client, serverului sau IED-ului/IED-urilor;

4. Stare inacceptabilă:

- majoritatea sau chiar toate componentele sistemului nu mai funcţionează sau nu funcţionează conform specificaţiilor;
- sistemul nu mai este disponibil;

Pentru estimarea stării curente a sistemului se pot folosi:

- informaţii din fişierele log;
- informaţii din bazele de date ale sistemului;
- informaţii privind resursele de sistem utilizate de către servere;

şi altele.

Metoda propusă de mine pentru estimarea stării curente şi pentru calculul timpilor de tranziţie între stări se bazează pe analiza fişierelor log.

În cazul sistemului de monitorizare şi control EMCSIT folosit ca studiu de caz şi prezentat în capitolul 4, fişierele log create de aplicaţiile server conţin informaţii privind data achiziţiei fiecărui pachet de date, id-ul pachetului, dacă acesta a fost valid şi starea echipamentului electric monitorizat.

Analizând fişierul log, se poate observa dacă au lipsit pachete de date din cele care trebuiau să fie achiziţionate la intervale de timp prestabilite (de ex. la fiecare minut).

Observând comportamentul sistemului pentru o perioadă suficientă de timp, am constatat că trecerea dintr-o stare în alta este strâns corelată cu numărul de pachete de date pierdute (neachiziţionate).

Astfel, propun următoarea clasificare a stărilor sistemului în cele 4 stări discrete:

- În cazul în care există mai puţin de 1 pachet de date ratat (neachiziţionat de către aplicaţia server) pe zi atunci sistemul se află în starea S1 (stare bună).
- Dacă există cel mult 1 pachet de date ratat la fiecare 6 ore sau în medie cel mult 4 pachete ratate pe zi atunci sistemul se află în starea S2 (stare acceptabilă). În aceeaşi stare sistemul se încadrează şi dacă are maximum două pachete de date ratate consecutiv.

- Dacă sunt cel mult 24 pachete ratate pe zi adică în medie 1 pachet de date la fiecare oră sau mai mult de 3 pachete de date ratate consecutiv atunci sistemul se află în starea S3 (stare proastă).
- Pentru situaţiile mai grave decât cele descrise anterior, sistemul se află în starea S4 (inacceptabilă).

Se poate utiliza modelul Markov pentru a stabili timpii de tranziţie între stările sistemului de monitorizare şi control şi respectiv când va ajunge sistemul în starea inacceptabilă.

Prin introducerea în modelul Markov al sistemului de monitorizare şi control, a două stări intermediare între cea bună şi cea inacceptabilă, este posibilă intervenţia promptă pentru remedierea defectelor înainte ca sistemul să ajungă în starea de funcţionare inacceptabilă.

În cazul în care o aplicaţie software se află în starea (S_i) se poate calcula FPT_{ij} adică timpul de tranziţie între starea S_i şi starea S_j.

Există mai multe metode de a calcula timpii de tranziţie FPT (First Passage Times). Una dintre metodele studiate este cea a combinării stărilor.

Această metodă combină toate stările intermediare (între starea iniţială i şi starea finală j) într-o singură stare (S) şi calculează rata de tranziţie din starea iniţială i în starea finală j notată FPTij (fig. 6.2.1.). Pentru aceasta se folosesc ratele de tranziţie λ_{iS} şi λ_{Sj}.

Ratele de tranziţie se calculează ca fiind numărul de cazuri când sistemul a trecut din starea i în starea j, raportat la numărul total de cazuri când şi-a schimbat starea.

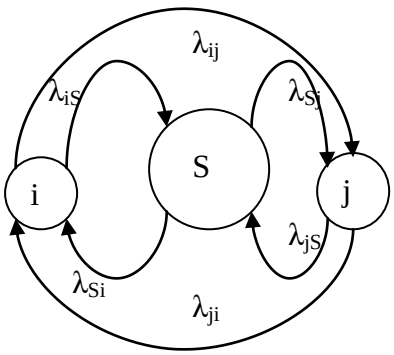

Fig. 6.2.1. Reprezentarea stărilor pentru calculul FPT

Ratele (λ_{ij}) şi (λ_{ji}) se obţin din calculele utilizând modelarea iniţială a stărilor, înainte de combinarea stărilor intermediare în starea S. Ratele de tranziţie în (S_i) din (S) se calculează cu metoda combinării stărilor:

(1) $$\lambda_{iS} = \sum_{z \in S} \lambda_{iz}$$

(2) $$\lambda_{Si} = \frac{\sum p_z \lambda_{zi}}{\sum_{z \in S} p_z}$$

(3) $$\lambda_{jS} = \sum_{z \in S} \lambda_{jz}$$

(4) $$\lambda_{Sj} = \frac{\sum p_z \lambda_{zj}}{\sum_{z \in S} p_z}$$

unde $z \in S$ este o stare, iar p_z este probabilitatea ca sistemul să fie în starea z.

Probabilitatea de a fi într-o anumită stare se calculează pentru acest sistem ca fiind numărul de cazuri în care sistemul software s-a aflat într-o anumită stare raportat la numărul total de cazuri.

Pentru calculul $FPT_{ij,}$ unde FPT_{ij} = (durata medie) timpul de tranziţie de la starea (i) la starea (j) se utilizează următoarea formulă:

(5) $$(FPT)_{ij} = \frac{\lambda_{Si} + \lambda_{Sj} + \lambda_{iS}}{\lambda_{Si}\lambda_{ij} + \lambda_{Sj}\lambda_{ij} + \lambda_{Sj}\lambda_{iS}}$$

După ce s-a calculat $(FPT)_{ij}$, rezultatul obţinut este exprimat în unitatea de timp pentru care au fost colectate datele. De ex. dacă se obţine$(FPT)_{ij}$ = 0,3 cu i=1 şi j=3 şi au fost luate în considerare date colectate pe un interval de 4 luni. Presupunând că sistemul software se află în starea S1 (stare bună) atunci conform modelului Markov, el va ajunge în starea S3 (stare proastă) în 0,3 luni.

Utilitatea calculării timpilor de tranziţie între stări a fost demonstrată în numeroase domenii printre care şi cel al ingineriei software. Considerând că o anumită aplicaţie software funcţionează fără probleme, pe baza informaţiilor privind ratele de defectare colectate în perioada de testare, se poate estima cu un anumit grad de încredere, momentul de timp aproximativ când sistemul va cădea.

6.3. Reţele Bayesiene

6.3.1. Modelul matematic

O reţea Bayesiană este un graf orientat fără cicluri ce reprezintă relaţiile probabilistice între variabilele unei mulţimi. Variabilele mulţimii sunt figurate drept noduri, notate cu $\{X_1,X_2,...X_n\}$ iar relaţiile între ele sunt reprezentate prin arce:

• nodul părinte reprezintă variabila care determină schimbarea variabilei fiu;

• nod fiu este variabila care suferă influenţa variabilei părinte;

Intensitatea şi modul în care se manifestă influenţele între variabile sunt redate prin tabelul de probabilităţi condiţionate al fiecărui nod în parte. Acesta ne permite să calculăm valorile probabilităţilor pentru variabila aleatoare asociată nodului, în funcţie de valorile părinţilor.

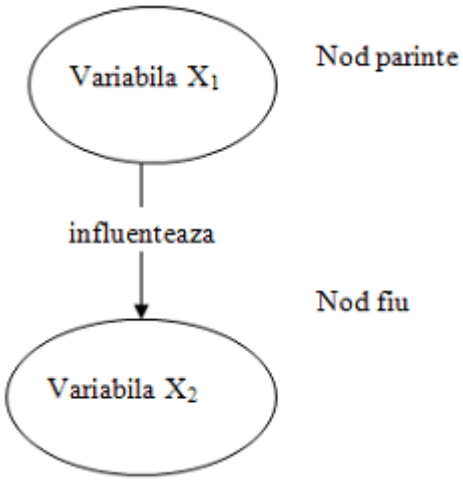

Fig. 6.3.1.1.Relaţia dintre nodul părinte şi nodul fiu într-o reţea Bayesiană

Semnificaţia intuitivă a arcului părinte-fiu este aceea că părintele influenţează în mod direct pe fiu.

Fiecare nod are un tabel de probabilităţi condiţionate asociat lui. Probabilităţile condiţionate sunt bazate pe informaţiile din trecut. O probabilitate condiţionată este scrisă matematic ca P(x|p1,p2,...pn) şi reprezintă probabilitatea ca variabila X să fie în starea x dacă părintele P1 se află în starea p1, P2 se află în starea p2,..., respectiv Pn se află în starea pn. Aceasta informaţie este reprezentată în tabela de probabilităţi condiţionate.

De exemplu, în figura care urmează se prezintă un exemplu privind influenţa precipitaţiilor asupra condiţiilor de drum ce poate fi interpretată în felul următor: „Dacă nu

sunt precipitaţii, atunci condiţiile de drum au o probabilitate de 5% să fie impracticabile şi de 95% să fie practicabile". În mod similar se pot citi restul de informaţii din tabela de probabilităţi condiţionate.

părinte	fiu	
precipitații	condiții de drum	
	impracticabil	practicabil
deloc	0,050	0,950
usoară	0,100	0,900
puternică	0,700	0,300

Fig. 6.3.1.2. Exemplu de tabelă de probabilităţi condiţionate

În cazul unei reţele Bayesiene, cum ar fi cea din fig. 6.3.1.3., pentru a calcula probabilitatea de defectare a nodurilor părinte notate cu Ai, se aplică formula lui Bayes pentru calculul probabilităţii condiţionate:

$$P(A_i|E = defect) = \frac{P(A_i) \times P(E = defect|A_i)}{\sum_{j=1}^{n} P(A_j) \times P(E = defect|A_j)}$$

unde n reprezintă nodurile ce influenţează direct nodul i, i=1,...,n.

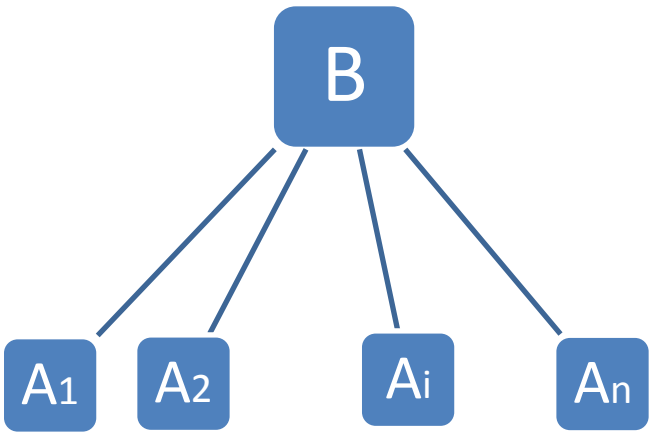

Fig. 6.3.1.3. Exemplu de reţea Bayesiană

Probabilitatea ca nodul fiu, B, să se defecteze este următoarea:

$$P(B) = \sum_{i=1}^{n} P(A_i) \times P(E = defect|A_i)$$

6.3.2. Modelarea unui sistem de monitorizare şi control a unei staţii electrice conform standardului IEC61850 printr-o reţea Bayesiană

Reţelele Bayesiene pot estima probabilitatea de defectare a unui sistem software pe baza istoricului ratelor de defectare ale modulelor constituente.

Ne propunem să modelăm printr-o reţea Bayesiană, un sistem de monitorizare şi control a unei staţii electrice [Urs8-6]. Aceasta ne va permite să calculăm probabilitatea de defectare a fiecărui nod din sistem precum şi a întregului sistem.

Pentru calculele referitoare la probabilităţile de defectare a nodurilor (IED-urilor) cât şi a sistemului de monitorizare şi control trebuie achiziţionate informaţii privind numărul de defecte descoperite într-o anumită perioadă de timp, pentru fiecare IED.

O astfel de reţea, care modelează un sistem de monitorizare şi control conform standardului IEC61850, este o reţea Bayesiană cu 2 nivele:

- Pe primul nivel (nivelul 0) se află nodul fiu (B) ce reprezintă sistemul software de monitorizare şi control a unei staţii electrice;
- Pe al doilea nivel (nivelul 1) se află nodurile părinte (Ai) respectiv serverele locale;

Defectarea unui IED va afecta într-o măsură mai mare sau mai mică defectarea întregului sistem de monitorizare şi control.

Fiecare nod fiu este caracterizat de o anumită rată de defectare (număr de defecte într-o perioadă de timp).

Reţeaua Bayesiană, fiind un graf orientat, suportă orice relaţie între noduri. În funcţie de relaţiile/influenţa între noduri, se calculează probabilitatea de defectare a acelui nod.

Sistemul de monitorizare şi control a staţiei electrice, fiind nod fiu este influenţat de probabilităţile de defectare a tuturor nodurilor.

În cazul în care nu toate nodurile influenţează sistemul, atunci formula lui Bayes se modifică corespunzător.

La numitor vor apărea informaţii doar despre acele noduri care influenţează nodul pentru care se calculează probabilitatea de defectare.

De exemplu, pentru următoarea situaţie:

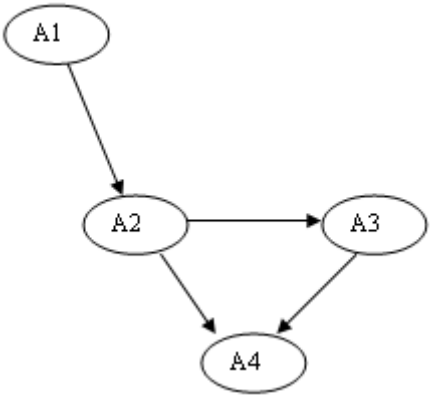

Fig. 6.3.2.1. Exemplu de influenţă între nodurile unei reţele Bayesiene

avem următoarele formule pentru calculul probabilităţii de defectare a nodurilor A2, A3 şi respectiv A4:

$$P(A_2|E = defect) = \frac{P(A_2) \times P(E = defect|A_2)}{P(A_1) \times P(E = defect|A_1) + P(A_2) \times P(E = defect|A_2)}$$

$$P(A_3|E = defect) = \frac{P(A_3) \times P(E = defect|A_3)}{P(A_2) \times P(E = defect|A_2) + P(A_3) \times P(E = defect|A_3)}$$

$$P(A_4|E = defect) = \frac{P(A_4) \times P(E = defect|A_4)}{P(A_2) \times P(E = defect|A_2) + P(A_3) \times P(E = defect|A_3) + P(A_4) \times P(E = defect|A_4)}$$

În figura 6.3.2.1., nodul A1 influenţează doar nodul A2, în timp ce acesta influenţează nodurile A3 şi A4. Nodul A4 este influenţat de către nodurile A2 şi A3.

De aceea, în formula ce calculează probabilitatea de defectare a nodului A2, la numitor sunt utilizate probabilităţile pentru nodurile A2 şi A1; în formula ce calculează probabilitatea de defectare a nodului A3, la numitor sunt utilizate probabilităţile pentru nodul A2 şi A3; în formula ce calculează probabilitatea de defectare a nodului A4, la numitor sunt utilizate probabilităţile pentru nodul A2, A3 şi respectiv A4.

Un sistem de monitorizare şi control ce respectă standardul IEC61850, în varianta simplificată are structura din următoarea figură:

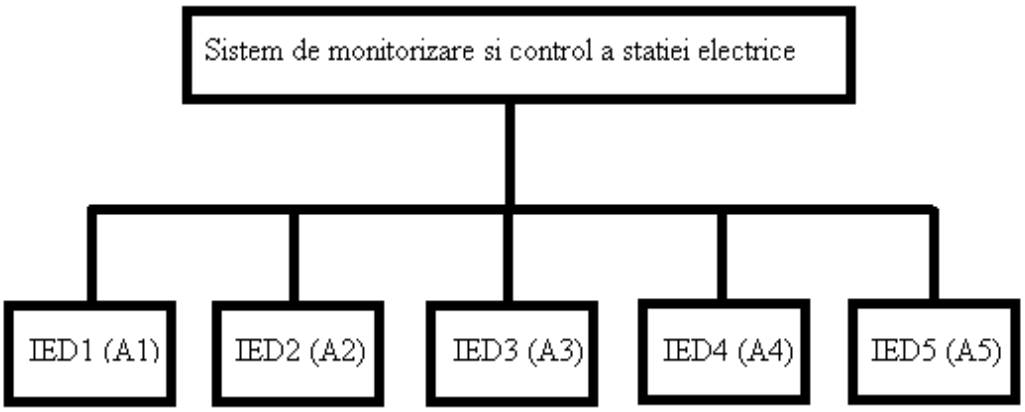

Fig. 6.3.2.2. Arhitectura simplificată a sistemului de monitorizare şi control

Fiecare nod de tip părinte IED monitorizează un singur tip de echipament electric aparţinând staţiei electrice. Sistemul de monitorizare şi control a staţiei electrice este nodul fiu şi este influenţat direct şi în egală măsură de toate nodurile IED.

6.3.3. Studiu de caz: modelarea sistemului EMCSIT printr-o reţea Bayesiană

Sistemul de monitorizare şi control EMCSIT este alcătuit din următoarele componente:

- IED-uri pentru fiecare echipament electric primar din staţie;
- Servere locale corespunzătoare mai multor IED-uri ce sunt grupate în celule electrice;
- Serverul central de baze de date;
- Calculatorul pe care se execută aplicaţia client, aflat în camera de comandă a staţiei electrice;

Sistemul EMCSIT poate fi modelat prin mai multe reţele Bayesiene. Astfel, se poate construi câte o reţea Bayesiană pentru fiecare server local, ce poate fi descris ca un subsistem de monitorizare şi control; vor fi 5 reţele Bayesiene cu ajutorul cărora se va putea calcula probabilitatea de defectare a fiecărui subsistem de monitorizare şi control. Informaţiile utilizate sunt cele referitoare la ratele de defectare/cădere pentru fiecare IED ce face parte din reţea.

Se poate construi o reţea Bayesiană pentru serverul central, ce va avea drept noduri părinte serverele locale din cabinele de relee, notate cu Server local CR1, Server local CR2, Server local CR3, Server local CR4 respectiv Server local CR5 adică subsistemele de

monitorizare şi control. Vor fi utilizate informaţiile privind probabilităţile de defectare ale serverelor locale, calculate anterior.

În final, reţeaua Bayesiană prin care se va putea calcula probabilitatea de defectare a întregului sistem de monitorizare şi control a staţiei electrice va avea drept noduri părinte: serverul central, IED-ul ce monitorizează transformatorul de putere (de tipul IED5), IED-ul ce monitorizează bobina de compensare (de tipul IED5) şi calculatorul ce rulează aplicaţia client din camera de comandă. Se vor folosi ca date de intrare, probabilitatea de defectare calculată anterior pentru serverul central şi ratele de defectare/cădere pentru IED-urile de tip IED5 şi aplicaţia client.

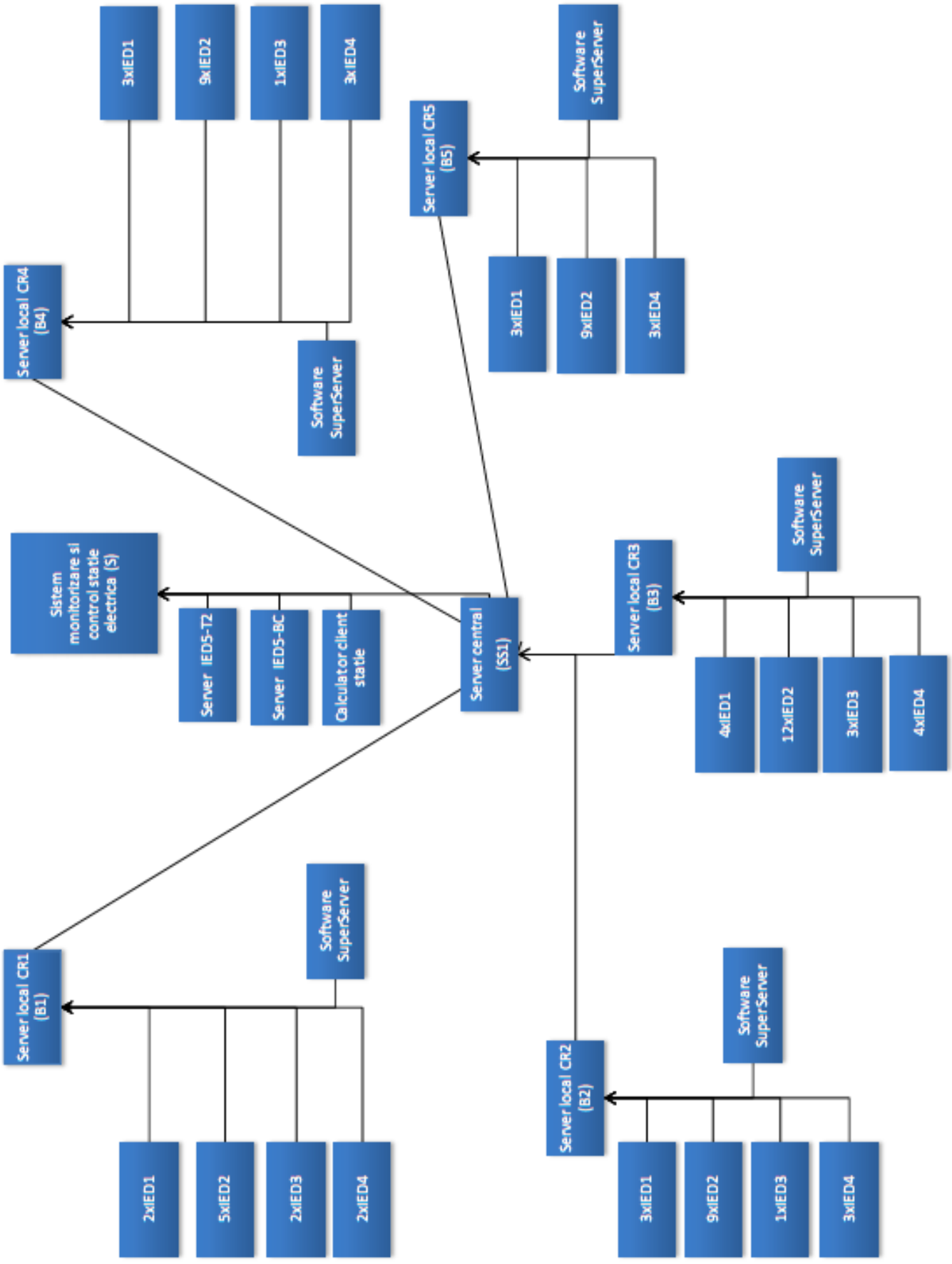

Fig. 6.3.3.1. Reprezentarea sistemului EMCSIT printr-o reţea Bayesiană

În cabina de relee 1 (CR1), sunt conectate la un server local IED-urile de tip IED1, IED2, IED3 şi IED4. Acest server va fi reprezentat în reţeaua Bayesiană ca fiind nodul B1.

Similar, în cabina de relee 2 (CR2), sunt conectate IED-urile la un server local, reprezentat în reţeaua Bayesiană ca fiind nodul B2; în cabina de relee 3 (CR3), sunt conectate IED-urile la un server local, reprezentat în reţeaua Bayesiană ca fiind nodul B3; în cabina de relee 4 (CR4), sunt conectate IED-urile la un server local, reprezentat în reţeaua Bayesiană ca fiind nodul B4; în cabina de relee 5 (CR5), sunt conectate IED-urile la un server local, reprezentat în reţeaua Bayesiană ca fiind nodul B5.

Se vor calcula probabilităţile de defectare ale nodurilor B1, B2, B3, B4 şi B5 în funcţie de ratele de defectare ale IED-urile care aparţin de respectivele cabine de relee.

Toate cele 5 noduri reprezentând serverele locale sunt conectate în reţea la un server central de baze de date denumit în reţeaua Bayesiană ca fiind nodul SS1.

Funcţionarea acestui nod va fi influenţată de nodurile B care sunt independente între ele.

În final, aplicaţia EMCSIT Staţie, ce rulează pe un calculator aflat în camera de comandă, va avea propriile rate de defectare: informaţii achiziţionate în timpul testelor efectuate în staţie cu toate IED-urile şi serverele conectate şi în operare.

Funcţionarea acestui nod, reprezentat în reţeaua Bayesiană ca fiind nodul S, va fi influenţată pe lângă funcţionarea software-ului ce rulează local şi de nodul SS1 deci indirect de nodurile B.

În continuare putem calcula probabilităţile de defectare pentru fiecare server local B1, B2, B3, B4 respectiv B5, conform formulei lui Bayes:

$$P(B_1|E = defect) = \frac{P(B_1) \times P(E = defect|B_1)}{P(A_1) \times P(E = defect|A_1) + P(A_2) \times P(E = defect|A_2) + P(A_3) \times P(E = defect|A_3) + P(A_4) \times P(E = defect|A_4)}$$

$$P(B_2|E = defect) = \frac{P(B_2) \times P(E = defect|B_2)}{P(A_1) \times P(E = defect|A_1) + P(A_2) \times P(E = defect|A_2) + P(A_3) \times P(E = defect|A_3) + P(A_4) \times P(E = defect|A_4)}$$

$$P(B_2|E = defect) = \frac{P(B_2) \times P(E = defect|B_2)}{P(A_1) \times P(E = defect|A_1) + P(A_2) \times P(E = defect|A_2) + P(A_3) \times P(E = defect|A_3) + P(A_4) \times P(E = defect|A_4)}$$

$$P(B_4|E = defect) = \frac{P(B_4) \times P(E = defect|B_4)}{P(A_1) \times P(E = defect|A_1) + P(A_2) \times P(E = defect|A_2) + P(A_3) \times P(E = defect|A_3) + P(A_4) \times P(E = defect|A_4)}$$

$$P(B_5|E = defect) = \frac{P(B_5) \times P(E = defect|B_5)}{P(A_1) \times P(E = defect|A_1) + P(A_2) \times P(E = defect|A_2) + P(A_3) \times P(E = defect|A_3) + P(A_4) \times P(E = defect|A_4)}$$

Apoi se poate calcula probabilitatea de defectare a nodului SS1:

$$P(SS_1|E = defect) = P(B_1) \times P(E = defect|B_1) + P(B_2) \times P(E = defect|B_2) + P(B_3) \times P(E = defect|B_3) + P(B_4) \times P(E = defect|B_4) + P(B_5) \times P(E = defect|B_5)$$

În final se calculează probabilitatea de defectare a întregului sistem, nodul fiu care este influenţat indirect de toate IED-urile prin intermediul serverelor locale şi serverului central de baze de date:

$$P(S|E = defect) = P(SS_1) \times P(E = defect|SS_1) + P(A_5T2) \times P(E = defect|A_5T2) + P(A_5BC) \times P(E = defect|A_5BC) + P(HMI) \times P(E = defect|HMI)$$

6.4. Concluzii

În acest capitol am prezentat aplicarea a trei modele matematice pentru estimarea fiabilităţii unui sistem de monitorizare şi control a unei staţii electrice: rata defectelor în timpul operării folosind modelul Rayleigh, perioada de timp în care sistemul poate ajunge într-o stare de funcţionare necorespunzătoare sau cădere folosind un lanţ Markov, probabilitatea de defectare în timpul operării folosind o reţea Bayesiană. Modelul Rayleigh a fost utilizat pentru aplicaţiile server corespunzătoare fiecărui tip de IED din cadrul sistemului EMCSIT, pentru a observa dacă rata descoperirii defectelor s-a stabilizat şi este mică la sfârşitul perioadei de dezvoltare, inclusiv a testelor în staţie (cu IED-urile conectate la echipamentele electrice), în caz contrar existând posibilitatea sa apară defecte în perioada imediat următoare.

Modelul Markov a fost folosit cu succes în domeniul energetic, pentru estimarea timpilor de tranziţie între stările diverselor echipamente electrice primare. Aceasta estimare se face în scopul evaluării momentului în care echipamentele vor ceda. Pentru aceasta, trebuie că pe baza parametrilor măsuraţi ai echipamentelor, să se facă o încadrare a stării curente a echipamentului. În proiectele din domeniul energetic [Urs8-20] s-a propus ca un echipament electric să fie modelat prin 4 stări posibile (în care se poate afla acesta), în funcţie de parametrii măsuraţi şi limitele tehnologice aplicate: stare bună, acceptabilă, proastă şi inacceptabilă. Am propus o modalitate similară de modelare a stărilor unui sistem software de monitorizare şi control a unei staţii electrice, în scopul utilizării modelului Markov pentru calculul FPT (first passage times). Utilizând aceasta metodă se poate calcula timpul în care sistemul poate ajunge în starea inacceptabilă.

De asemenea, am propus modelarea unui sistem de monitorizare şi control a unei staţii electrice utilizând reţelele Bayesiene pentru calculul probabilităţii de defectare a întregului sistem. Se poate calcula probabilitatea de defectare a fiecărui nod al reţelei în funcţie de influenţa altor noduri asupra sa.

7. Aplicaţia software pentru estimarea fiabilităţii unui sistem de monitorizare şi control a unei staţii electrice

Aplicaţia software permite modelarea unui sistem de monitorizare şi control a unei staţii electrice printr-o reţea Bayesiană şi evaluarea reţelei în scopul estimării probabilităţii de defectare a fiecărui nod al reţelei precum şi a întregului sistem.

Configuraţia staţiei trebuie să fie descrisă într-un fişier în format SCL, conform standardului IEC61850. Fişierul poate fi creat cu un editor special (comercial) sau manual.

Aplicaţia permite:

- Preluarea configuraţiei unei staţii electrice dintr-un fişier scris în limbajul SCL;
- Adăugarea de noi noduri în configuraţie;
- Editarea reţelei Bayesiene prin definirea relaţiilor între noduri;
- Adăugarea sau calcularea de probabilităţi de defectare a priori pentru fiecare nod, pe baza ratelor de defectare;
- Calculul probabilităţii de defectare a fiecărui nod şi a întregului sistem ţinând cont de relaţiile existente între noduri în reţeaua Bayesiană;

Pentru construirea reţelei Bayesiene se efectuează următorii paşi:

- se creează modelul reţelei Bayesiene bazat pe structura staţiei electrice definită în fişierul SCL. Nodurile reţelei sunt IED-urile sau serverele locale la care se conectează IED-urile, pe care se execută aplicaţiile software tip server;
- se definesc tipurile de noduri: noduri de tip părinte şi fiu;
- se completează informaţiile privind ratele de defectare ale IED-urilor (numărul de defecte observate într-o anumită perioadă de timp) şi ale serverelor locale;

Aplicaţia a fost realizată în mediul de dezvoltare Microsoft Visual Studio .NET 2010. Este compusă din modulele *Parser SCL* şi *Calcul BN*.

7.1. Modulul Parser SCL

Pentru modelarea unei staţii electrice conform standardului IEC61850 s-a folosit limbajul SCL (Substation Configuration Language) şi s-a utilizat aplicaţia Visual SCL [Iec9-11].

Aplicaţia Visual SCL permite utilizatorului să salveze schemele de staţie în formatul standard tip XML specific IEC61850 sub forma unor fişiere cu extensia .SCD.

Pentru utilizarea informaţiilor prezente în fişierele .SCD am dezvoltat un modul software, denumit *Parser SCL*. Acesta citeşte informaţii din fişierele .SCD scrise în limbajul de configurare a unei staţii electrice – SCL (IEC61850) şi furnizează ca date de ieşire, structuri de date ce pot fi apoi definite ca noduri ale unei reţele Bayesiene. Datele de ieşire sunt folosite ulterior de aplicaţia *Calcul BN*, pentru estimarea fiabilităţii sistemului.

Nodurile şi informaţiile asociate lor, reprezentând datele de defectare ale IED-urilor, vor folosi la calculele privind probabilitatea de defectare a sistemului de monitorizare şi control.

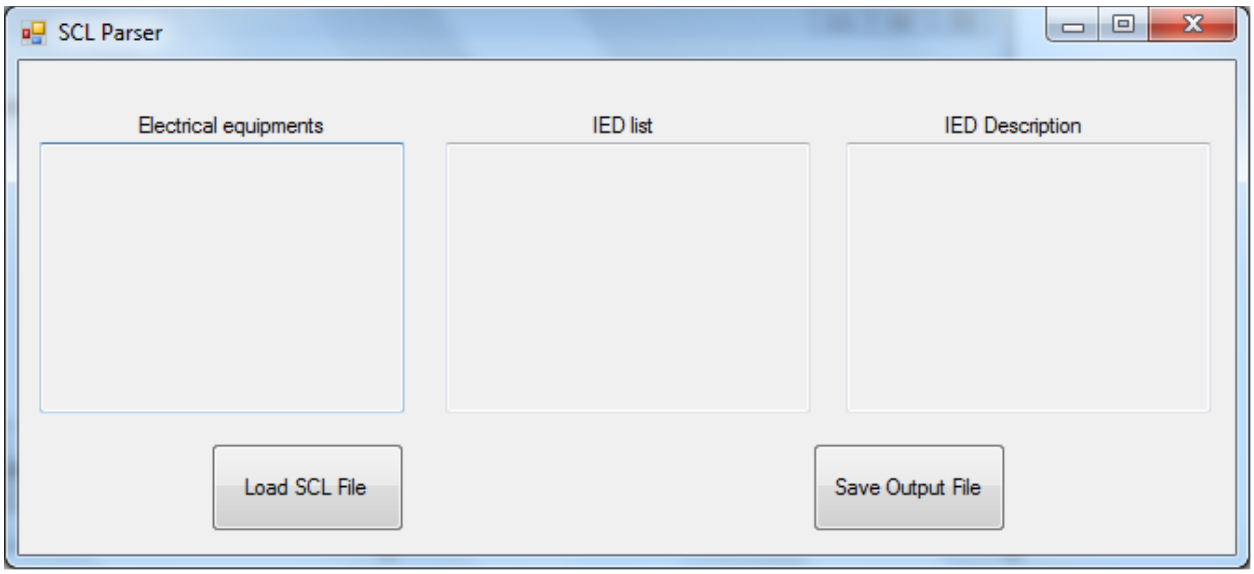

Fig. 7.1.1. Fereastra principală a modulului *Parser SCL*.

În cadrul ferestrei principale, utilizatorul are opţiunea de a încărca fişierul SCL dorit. Informaţiile obţinute din acest fişier sunt vizibile în cele 3 secţiuni:

- Electrical equipments (echipamentele electrice din staţie);
- IED list (IED-urile asociate echipamentelor electrice);
- IED description (descrierea IED-urilor);

Informaţiile pot fi salvate într-un fişier extern în format text prin apăsarea butonului "Save Output File".

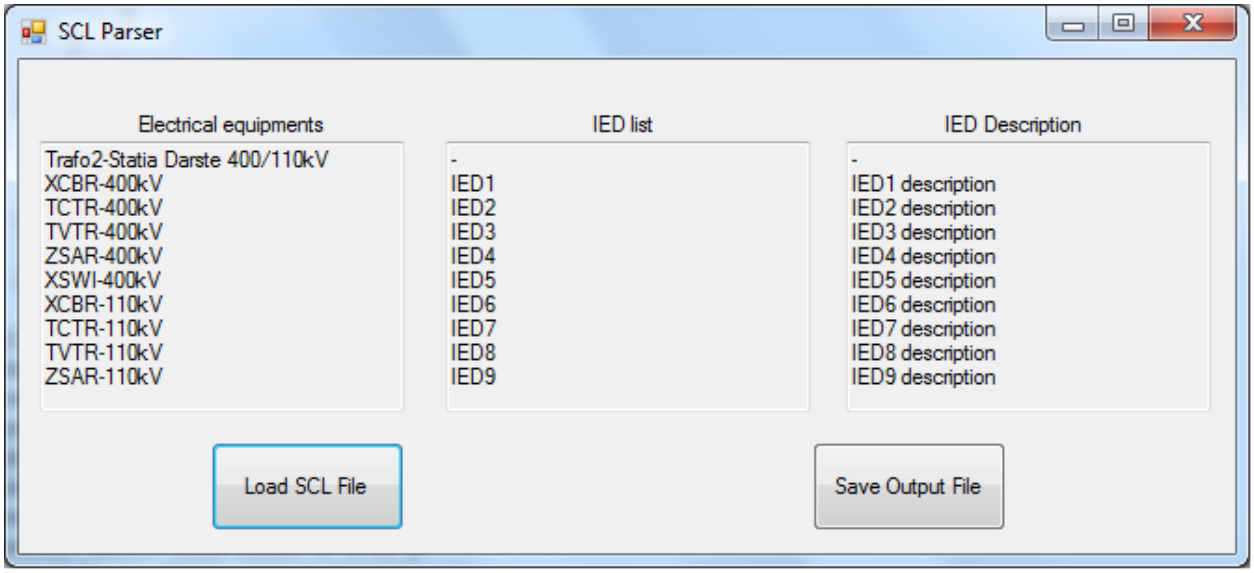

Fig. 7.1.2. Rezultate obţinute

Modulul *Parser SCL* a fost implementat iniţial ca o aplicaţie independentă iar ulterior a fost integrat în aplicaţia curentă. Salvarea datelor de ieşire ale modulului Parser SCL este optională, ele fiind puse direct la dispoziţia aplicaţiei curente.

7.2. Modulul Calcul BN

Informaţiile obţinute, respectiv echipamentele electrice şi IED-urile asociate se folosesc în modelarea matematică a reţelei Bayesiene ca noduri ale sistemului de monitorizare şi control a staţiei electrice.

După ce au fost făcute legăturile între IED-uri şi echipamentele electrice, aplicaţia de calcul a reţelei Bayesiene preia lista de IED-uri şi oferă utilizatorului posibilitatea completării ratelor de defectare pentru acestea.

Suplimentar, se pot face conexiuni între IED-uri şi noduri introduse manual de utilizator, pentru a permite alcătuirea unei reţele Bayesiene cât mai apropiată de configuraţia fizică a sistemului de monitorizare şi control.

Fluxul completării informaţiilor în cadrul aplicaţiei software de modelare a reţelei Bayesiene este prezentat în următoarele figuri:

1. Se rulează modulul software *Parser SCL* ce va furniza o parte din datele de intrare ale aplicaţiei. Acestea sunt nodurile reţelei Bayesiene bazată pe schema staţiei electrice conform IEC61850.

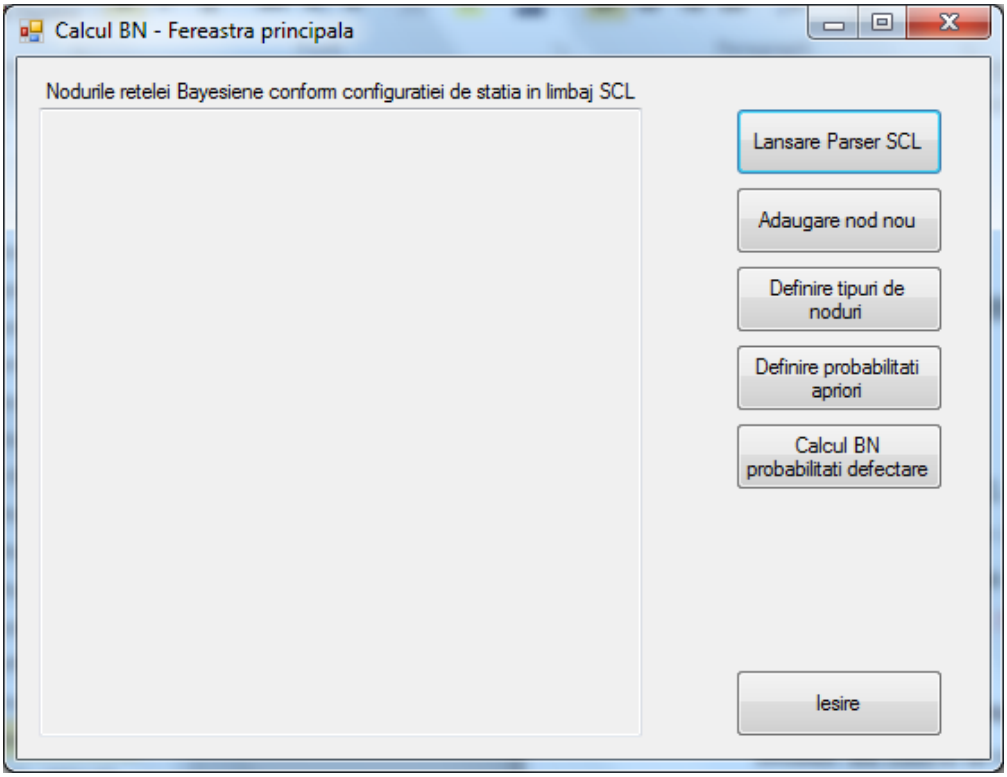

Fig. 7.2.1. Fereastra principală a aplicaţiei

După încărcarea informaţiilor din fişierul în format SCL, nodurile reţelei Bayesiene se vor regăsi în fereastra principală a aplicaţiei *Calcul BN*, în partea stângă. Sunt următoarele tipuri de noduri:

- Noduri reprezentând IED-urile ce monitorizează echipamentele electrice din staţie;
- Noduri de tip PC, definite de utilizator, ce reprezintă serverele locale din cabinele de relee şi respectiv serverul central de baze de date;
- Nodul de tip HMI (Human Machine Interface) ce reprezintă calculatorul client din camera de comandă ce rulează aplicaţia software tip client;

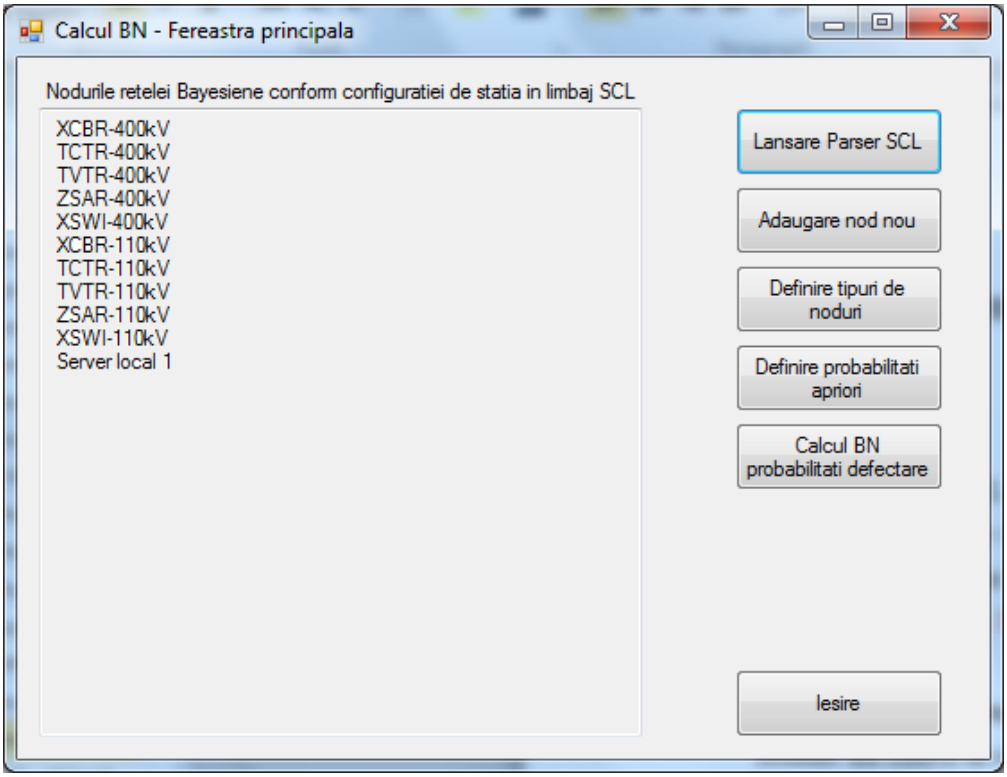

Fig. 7.2.2. Nodurile reţelei Bayesiene

2. Există posibilitatea adăugării de noi noduri pentru reţeaua Bayesiană, pe lângă cele obţinute din fişierul de configurare a staţiei electrice, conform standardului IEC61850.

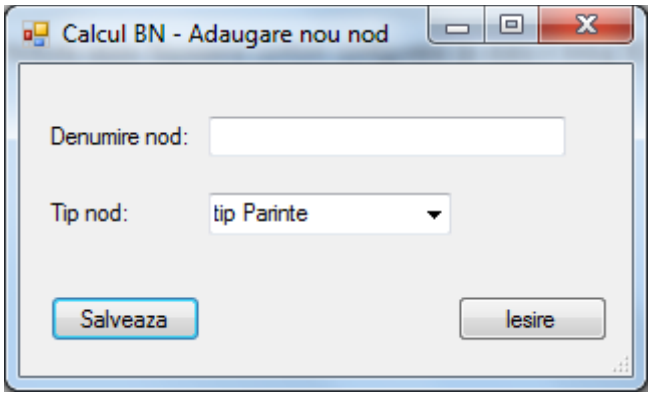

Fig. 7.2.3. Adăugarea de noi noduri în reţeaua Bayesiană

3. Sunt definite nodurile de tip părinte (IED-urile) şi nodul fiu (subsistemul de monitorizare şi control a staţiei electrice - serverul local):

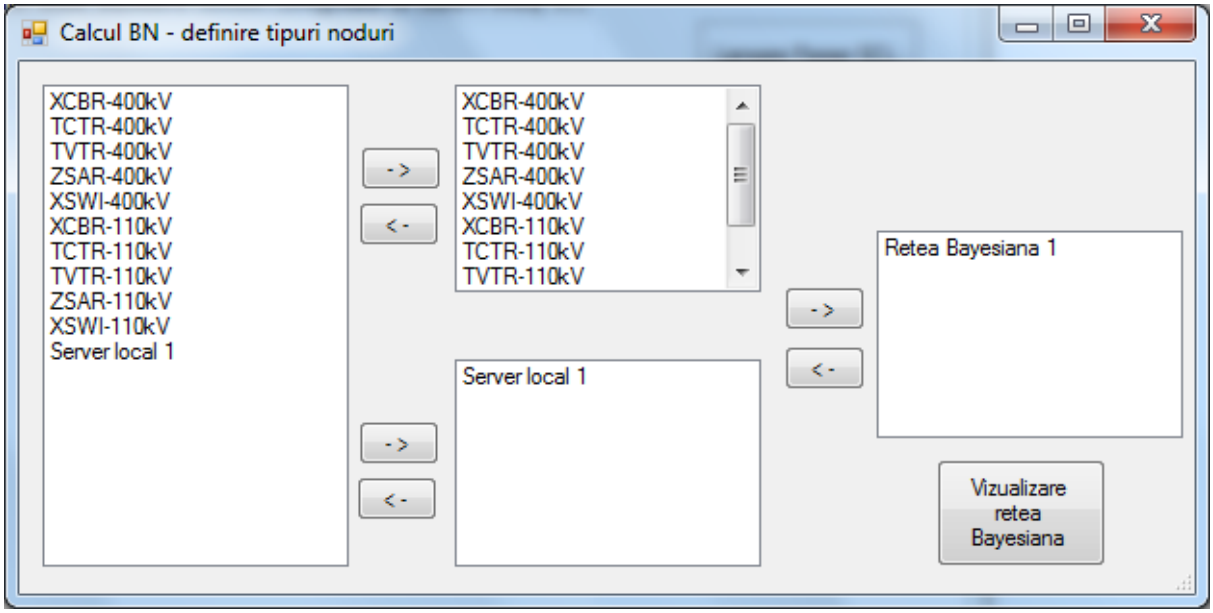

Fig. 7.2.4. Definirea tipurilor de noduri pentru reţeaua Bayesiană.

După definirea tipurilor de noduri, se pot vizualiza separat aceste informaţii într-o noua fereastră:

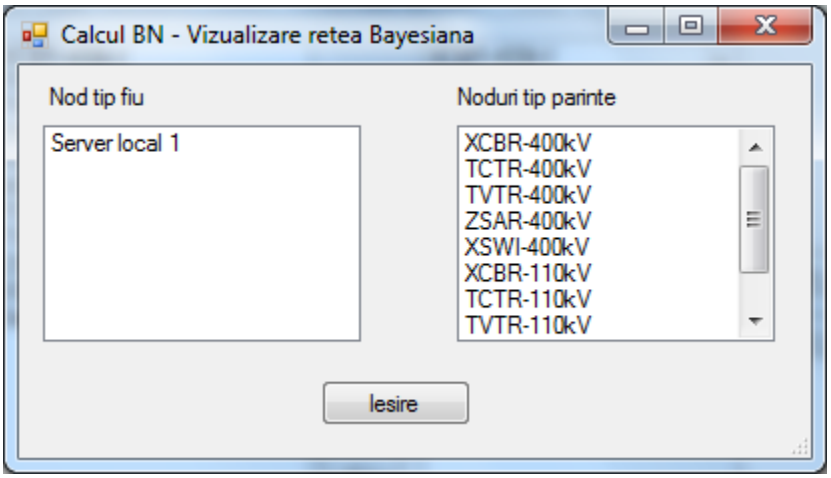

Fig. 7.2.5. Vizualizare noduri reţea Bayesiană şi tipul lor

4. Pentru calculul fiabilităţii fiecărui IED, se încarcă informaţii privind numărul de defecte constatate de-a lungul perioadei de testare (rata de defectare), pentru fiecare nod al reţelei Bayesiene – IED_1,..., IED_n.

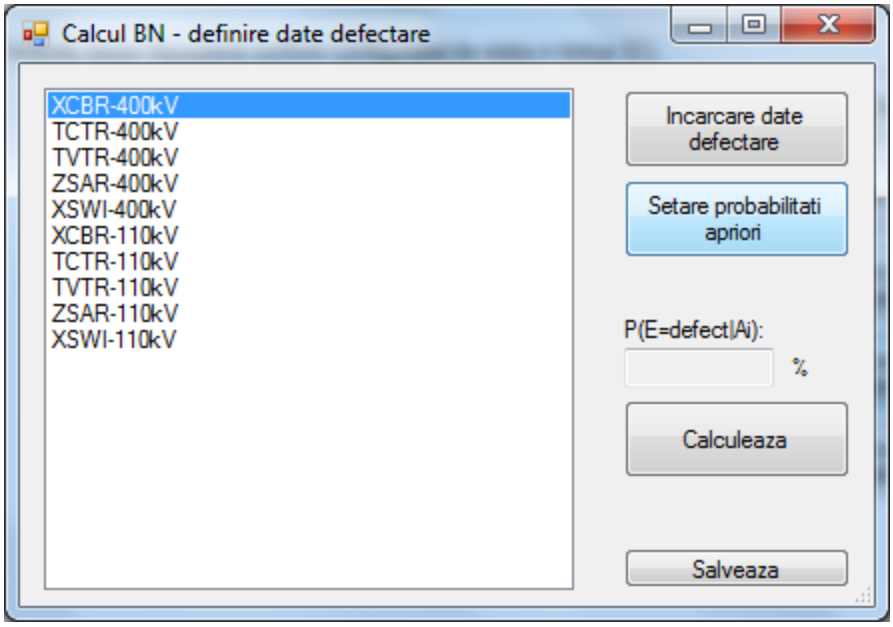

Fig. 7.2.6. Fereastra de încărcare a datelor de defectare pentru IED-uri.

În situaţia în care nu sunt disponibile date privind defectele înregistrate pentru un anumit IED, se poate seta manual probabilitatea a priori de defectare.

5. După ce au fost completate toate informaţiile prezentate anterior, aplicaţia calculează probabilitatea de defectare a sistemului de monitorizare şi control precum şi probabilitatea de defectare a fiecărui IED dar şi a subsistemelor (în cazul în care au fost definite). În studiul de caz, aceste subsisteme sunt reprezentate de serverele locale din cabinele de relee.

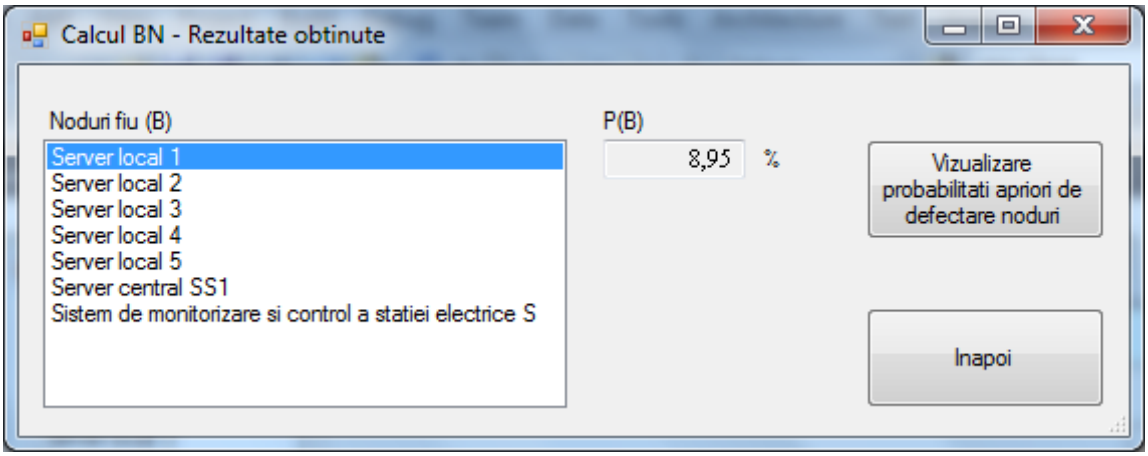

Fig. 7.2.7. Fereastra de afişare a rezultatelor finale

7.3. Rezultate obţinute

Pentru estimarea fiabilităţii unui sistem de monitorizare şi control utilizând metoda reţelelor Bayesiene, a fost folosită o configuraţie de staţie de transformare aparţinând reţelei de transport a energiei electrice din România, ce include 5 tipuri de IED-uri câte unul pentru fiecare tip de echipament electric primar:

- IED1 pentru întreruptor;
- IED2 pentru separator;
- IED3 pentru descărcător;
- IED4 pentru transformator de măsură de curent/tensiune;
- IED5 pentru transformator de putere/bobină de compensare;

Fişierul ce conţine schema completă a staţiei de 400/110kV folosită pentru calcule în studiul de caz, se regăseşte în anexa 2.

Pentru implementarea practică a modelului matematic al reţelei Bayesiene, am făcut calculele pe substaţia de 110kV care are în componenţă următoarele IED-uri descrise în fişierul SCL:

- 15 IED-uri de tip IED1 (notate cu A1);
- 44 IED-uri de tip IED2 (notate cu A2);
- 7 IED-uri de tip IED3 (notate cu A3);
- 15 IED-uri de tip IED4 (notate cu A4);

În total în substaţia de 110kV sunt 81 de IED-uri ce monitorizează echipamentele electrice primare.

Faptul că am luat drept exemplu o parte din toată staţia electrică, nu aduce modificări în teoria modelării unei întregi staţii electrice, fie că aparţine reţelei de distribuţie, fie că aparţine reţelei de transport, fie că este staţie de evacuare sau orice alt tip de staţie electrică.

Sunt aceleaşi categorii de echipamente electrice primare, diferă doar numărul şi poziţionarea lor în cadrul staţiei electrice.

În literatura de specialitate [Ban9-7] se foloseşte următoarea notaţie pentru calculul probabilităţii de defectare a unui sistem format din nodurile Ai:

$$P(E = defect) = \sum_{i=1}^{n} P(A_i) \times P(E = defect \mid A_i)$$

Evenimentul E este un eveniment oarecare pentru care modelăm reţeaua Bayesiană, în cazul nostru ne interesează probabilitatea de defectare a întreg ansamblului de IED-uri ce

constituie sistemul de monitorizare şi control a unei staţii electrice. Deci E reprezintă evenimentul de defectare a sistemului.

Pentru calculul probabilităţii de defectare, trebuie calculată pentru fiecare nod Ai (IEDi) probabilitatea de defectare a priori. Aceste probabilităţi se obţin, colectând date din timpul testării IED-urilor. Acestea conţin informaţii privind numărul de căderi sau defectări al IED-ului într-o anumită lună.

Numărul de căderi ale IED-urilor repartizate pe cabinele de relee este prezentat în următorul tabel:

Tabel 7.3.1. Număr de căderi pe tipuri de IED-uri pentru fiecare cabină de relee (CR)

Tip IED	**CR1**	**CR2**	**CR3**	**CR4**	**CR5**
IED1	2	0	4	0	0
IED2	1	3	12	0	3
IED3	0	1	4	0	0
IED4	0	0	4	0	1

Numărul de defecte prezentat în tabelul 7.3.1. reprezintă informaţii colectate pe o perioadă de 4 luni de teste cu IED-urile conectate la echipamentele electrice monitorizate, în staţia electrică.

S-au calculat probabilităţile de defectare pentru fiecare grup de IED-uri din cabinele de relee, unde acestea sunt conectate la câte un server local. În total sunt 5 astfel de servere locale.

De exemplu, pentru cabina de relee 1, se calculează probabilitatea de defectare a subsistemului de monitorizare şi control aferent cabinei de relee 1: **Server local CR1.**

În total sunt 11 IED-uri în cabina de relee 1: 2 IED-uri de tip IED1 (A1), 5 IED-uri de tip IED2 (A2), 2 IED-uri de tip IED3 (A3) şi 2 IED-uri de tip IED4 (A4). Utilizând aceste informaţii, se calculează probabilităţile pentru nodurile de tip IED din cabina de relee 1:

P(A1) = 2/11 = 0,18

P(A2) = 5/11 = 0,46

P(A3)= 2/11 = 0,18

P(A4) = 2/11 = 0,18

Utilizând informaţiile din următorul tabel:

Tabel A4.1. Număr de căderi pentru fiecare IED, în perioada de 4 luni de test în staţia electrică

IED1	**IED2**	**IED3**	**IED4**
2 căderi / 2 IED-uri diferite	1 cădere / 1 IED	0 căderi	0 căderi

se calculează probabilităţile de defectare a priori ale IED-urilor din cabina de relee 1.

Pentru IED-urile de tip IED1 din cabina de relee 1, au fost constatate 2 căderi la 2 IED-uri diferite de tip IED1, în perioada de 4 luni.

P(E=defect|A1) = 0,25

Pentru IED-urile de tip IED2 din cabina de relee 1, a fost constatată 1 cădere la 1 singur IED din totalul de 5 IED-uri de tip IED2, în perioada de 4 luni.

P(E=defect|A2) = 0,05

Pentru IED-urile de tip IED3 din cabina de relee 1, n-a fost constatată nici o cădere în perioada de 4 luni.

P(E=defect|A3) = 0

Pentru IED-urile de tip IED4 din cabina de relee 1, n-a fost constatată nici o cădere în perioada de 4 luni.

P(E=defect|A4) = 0

Se calculează probabilitatea de defectare a nodului tip server local din cabina de relee 1, notat în continuare CR1, modelat prin reţeaua Bayesiană:

P(CR1) = P(A1) x P(E=defect|A1) + P(A2) x P(E=defect|A2) + P(A3) x P(E=defect|A3) + P(A4) x P(E=defect|A4) = 0,25 x 0,18 + 0,05 x 0,46 = 0,068

Deci, probabilitatea de defectare a subsistemului de monitorizare din cabina de relee 1 este de 6,8%.

Calculele detaliate se regăsesc în Anexa 4. Rezultatele obţinute sunt următoarele:

- probabilitatea de defectare a subsistemului de monitorizare corespunzător cabinei de relee 2, P(CR2) este 5,2%.
- probabilitatea de defectare a subsistemului de monitorizare corespunzător cabinei de relee 3, P(CR3) este 26%.

- probabilitatea de defectare a subsistemului de monitorizare corespunzător cabinei de relee 4, P(CR4) este 0%.
- probabilitatea de defectare a subsistemului de monitorizare corespunzător cabinei de relee 5, P(CR5) este 6,6%.

Pentru calculul probabilităţii de defectare a nodului de tip server central, notat în continuare SS1, se folosesc probabilităţile de defectare ale serverelor locale CR1, CR2, CR3, CR4 şi CR5, calculate anterior.

Se aplică formula lui Bayes pentru calculul probabilităţii de defectare a nodului SS1:

P(E=defect) = 0,2 x 0,068 + 0,2 x 0,052 +0,2 x 0,26 +0,2 x 0 +0,2 x 0,066 = 0,0895

Modelând sistemul de monitorizare şi control din studiul de caz obţinut prin citirea informaţiilor din fişierul de configurare a staţiei electrice în limbajul SCL, prin 5 reţele Bayesiene (aferente cabinelor de relee, formate din noduri de tip IED), se obţine pentru acest sistem o **probabilitate de defectare de 8,95%.**

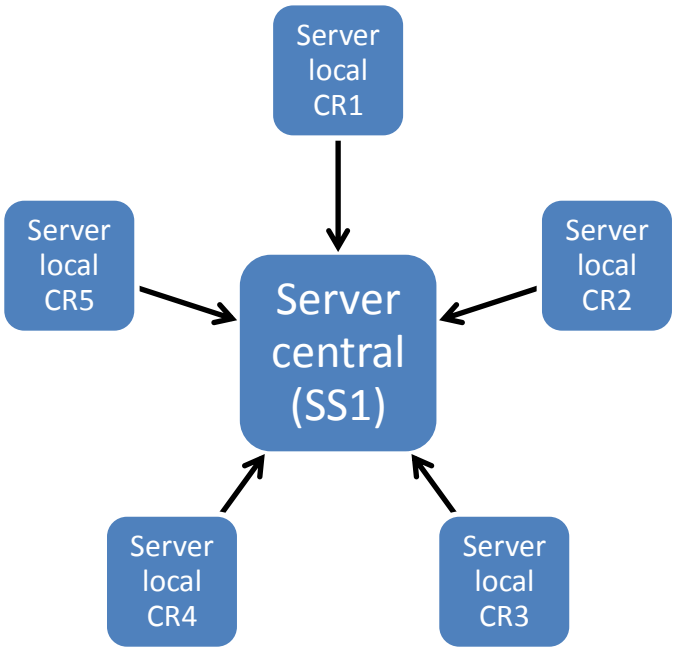

Fig. 7.3.1. Reţeaua Bayesiană pentru serverul central SS1 şi serverele locale din cabinele de relee

Putem completa configuraţia staţiei electrice în limbajul SCL conform standardului IEC61850, adăugând noduri de tip server şi noduri ce rulează aplicaţii software.

Astfel vom extinde modelarea prin reţele Bayesiene a sistemului de monitorizare şi control, adăugând noduri pentru serverele locale aferente IED5 şi pentru calculatorul din camera de comandă ce rulează aplicaţia tip client de monitorizare şi control.

Vom nota cu S (sistem de monitorizare) nodul fiu în această nouă reţea Bayesiană. Acesta este influenţat de nodul SS1 (serverul central de BD) a cărei probabilitate de defectare am calculat-o anterior, de două noduri de tip server IED5 (Trafomon Trafo2 – pentru monitorizarea transformatorului de mare putere respectiv Trafomon BC – pentru monitorizarea bobinei de compensare) precum şi de nodul ce reprezintă calculatorul din camera de comandă şi pe care rulează aplicaţia tip client. Acesta din urmă va fi notat cu HMI (Human Machine Interface) conform notaţiilor utilizate în cadrul standardului IEC61850.

În formula lui Bayes de calcul a probabilităţii de defectare a nodului S, sunt folosite următoarele informaţii:

- Probabilitatea de defectare a priori pentru nodul SS1 (calculată anterior), notată P(E=defect|SS1);
- Probabilitatea de defectare a priori pentru nodul IED5 corespunzător Trafomon-T2, notat P(E=defect|A5-T2);
- Probabilitatea de defectare a priori pentru nodul IED5 corespunzător Trafomon-BC, notat P(E=defect|A5-BC);
- Probabilitatea de defectare a priori pentru nodul HMI, notata P(E=defect|HMI);

Pentru calculatorul client din camera de comandă, notat HMI există următoarele informaţii din perioada de teste:

Tabel 7.3.2. Tabel număr defecte aplicaţie software client

Unitate de timp (lună)	**Numărul de căderi ale aplicaţiei software client**
4 (februarie)	0
5 (martie)	0
6 (aprilie)	0
7 (mai)	0

8 (iunie)	0
9 (iulie)	5
10 (august)	2
11 (septembrie)	0

Astfel, P(E=defect|HMI) = 2/8 = 0,25

Pentru nodul A5 (IED5) aferent Trafomon-T2 nu a existat nici o cădere în timpul celor 4 luni, P(E=defect|A5-T2) = 0.

Similar pentru nodul A5 aferent Trafomon-BC, P(E=defect|A5-BC) = 0.

Pentru nodul SS1 se va folosi rezultatul obţinut în urma calculelor anterioare utilizând reţeaua Bayesiană formată din nodurile părinte B1, B2, B3, B4 respectiv B5 corespunzătoare serverelor locale din cabinele de relee.

Se cunosc probabilităţile pentru nodurile reţelei Bayesiene:
P(SS1) = 0,5 deoarece sunt două servere centrale;
P(A5-T2) = 0,5 şi P(A5-BC) = 0,5 deoarece sunt două noduri de tip IED5;
P(HMI) = 1;

Se obţine: P(E=defect) = 0,5 x 0,0895 +0,5 x 0 + 0,5 x 0 + 1 x 0,25 = 0,2947

Probabilitatea de defectare a sistemului de monitorizare şi control S, modelat prin reţeaua Bayesiană ce include serverul central de baze de date SS1, cele două servere locale pentru IED-urile de tip IED5 şi calculatorul client din camera de comandă, **este de 29,47%.**

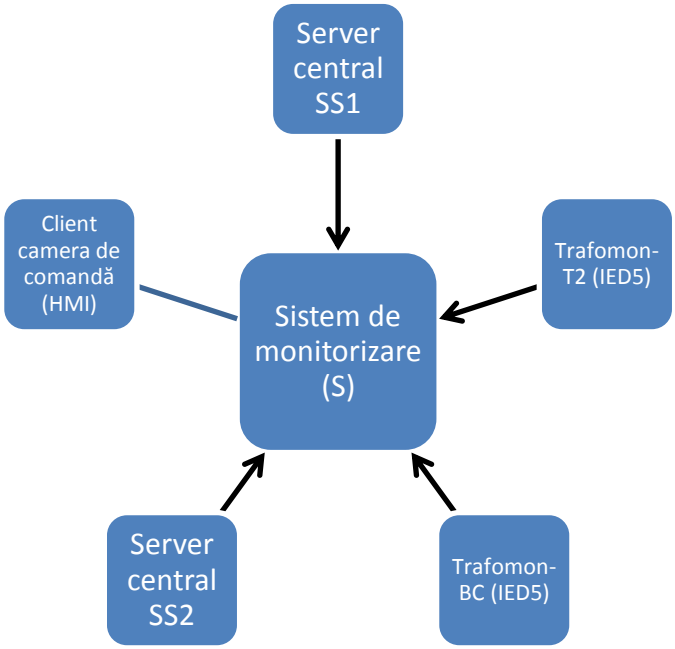

Fig. 7.3.2. Reţeaua Bayesiană pentru serverul central SS1 şi serverele locale din cabinele de relee

Deoarece am avut la dispoziţie informaţii privind căderile IED-urile din substaţia de 110kV, calculele de mai sus au fost făcute pentru un singur server central de baze de date, aferent substaţiei de 110kV. Pentru serverul central de baze de date aferent substaţiei de 400kV, serverele locale din cabinele de relee şi IED-urile aferente, nu am avut la dispoziţie suficiente informaţii pentru a efectua calculele.

Se poate calcula probabilitatea de defectare a sistemului pornind de la ipoteza că ambele servere au aceeaşi probabilitate de defectare. Se obţine următorul rezultat:

P(E=defect) = 0,5 x 0,0895 + 0,5 x 0,0895 + 0,5 x 0 + 0,5 x 0 + 1 x 0,25 = 0,3395

În acest caz, probabilitatea de defectare a sistemului de monitorizare şi control S este de 33,95%.

7.4. Concluzii

În acest capitol am prezentat aplicaţia software pe care am dezvoltat-o în scopul evaluării probabilităţii de defectare, în timpul operării, a unui sistem de monitorizare şi control a unei staţii electrice, folosind o reţea Bayesiană.

Configuraţia staţiei electrice include atât noduri de tip IED cât şi noduri de tip echipament electric. Fişierele în format SCL ce descriu configuraţia staţiei electrice,

precizează pentru fiecare echipament electric, IED-ul ce monitorizează parametrii acestuia. Întreg ansamblul de IED-uri ce monitorizează echipamentele electrice din staţie, alcătuiesc sistemul de monitorizare şi control.

Modelarea reţelei Bayesiene include nodurile de tip IED şi date privind rata de defectare a acestora în timpul testelor efectuate în staţie. Utilizând aceste informaţii, se poate estima fiabilitatea sistemului.

Am folosit aplicația pentru estimarea probabilității de defectare a sistemului din studiul de caz, folosind datele colectate pe parcursul dezvoltării şi exploatării sistemului.

Aplicația este alcătuită din modulul *Parser SCL* şi modulul *Calcul BN*. Modulul *Parser SCL* citeşte fişierele în format SCL şi apoi salvează informaţiile obţinute sub forma unor structuri de date reprezentând IED-urile şi echipamentele electrice monitorizate de către acestea. Modulul *Calcul BN* preia aceste date şi alcătuieşte nodurile unei reţele Bayesiene. Există posibilitatea completării reţelei Bayesiene cu noduri de tip server local, server central sau noduri ce rulează aplicaţii software.

După ce sunt definite nodurile tip părinte şi nodul tip fiu pentru o reţea Bayesiană, se introduc date privind ratele de defectare. În final, sunt efectuate calcule privind probabilitatea de defectare a serverelor locale, a serverului central de baze de date şi apoi a întregului sistem de monitorizare şi control a staţiei electrice. Rezultatele pot fi interpretate pentru a estima fiabilitatea sistemului.

8. Concluzii, contribuţii proprii şi planuri de cercetare pentru viitor

8.1. Concluzii

În cadrul tezei de doctorat am identificat principalele probleme ce afectează fiabilitatea sistemelor de monitorizare şi control a staţiilor electrice şi am propus soluţii de asigurare a calităţii acestor sisteme.

- În capitolul 2 am făcut o analiză a aspectelor de calitate software plecând de la un punct de vedere general, particularizând apoi pentru un sistem software de monitorizare şi control a unei staţii electrice. Astfel, sunt prezentate soluţii generale de asigurare a calităţii software precum: tehnici şi metode de prevenire a injectării defectelor, tehnici de eliminare a defectelor şi de izolare a acestora. S-a pus accentul pe activităţile de inspecţie şi testare, utilizarea blocurilor de recuperare, N-version programming, self-checking precum şi tehnici mai noi ca reconfigurarea şi reîntinerirea.

- În capitolul 3 au fost prezentate pe scurt câteva sisteme de monitorizare şi control, dezvoltate de cele mai reprezentative companii din domeniu, precum şi tehnologia Smart Grid. Au fost identificate problemele care pot apărea în dezvoltarea, funcţionarea şi operarea unui sistem de monitorizare şi control a unei staţii electrice, precum şi efectele acestora.

- În capitolul 4 am propus soluţii concrete pentru asigurarea calităţii unui sistem de monitorizare şi control a unei staţii electrice, folosind ca studiu de caz un sistem dezvoltat şi instalat într-o staţie electrică de transformare din România. Este prezentată arhitectura şi funcţionalitatea sistemului, denumit EMCSIT (Echipament pentru Monitorizarea Complexă a Staţiilor de Înaltă Tensiune), la a cărui dezvoltare am participat. Sunt descrise soluţiile şi exemple de aplicare a acestor soluţii pentru asigurarea calităţii acestui sistem. Pentru îmbunătăţirea procesului de testare a aplicaţiilor (EMCSIT Server) care primesc date în timp real de la dispozitive tip IED (Intelligent Electronic Device), am dezvoltat un simulator software de IED-uri care generează pachete de date şi le trimite pe interfaţa serială a serverului la care sunt cuplate. Pachetele de date sunt generate ţinând cont de domeniile de valori ale mărimilor monitorizate de fiecare tip de IED, aplicând

metoda *Pairwise testing*. În acest fel, se generează mult mai puţine cazuri de test faţă de numărul total al cazurilor care ar rezulta considerând toate combinaţiile posibile între valorile de test selectate pentru mărimile monitorizate, însă se pot descoperi aproximativ 70% din erori în timpul testării.

- În capitolul 5 sunt prezentate aspecte generale privind standardul IEC61850 şi utilizarea acestuia în cadrul unei staţii electrice. Standardul prevede utilizarea de IED-uri pentru monitorizarea echipamentelor electrice primare. Pentru descrierea configuraţiei unei staţii electrice, în cadrul standardului este definit limbajul SCL. Structura generală a unui fişier SCL permite descrierea echipamentelor electrice din cadrul unei staţii precum şi asocierea IED-urilor destinate monitorizării şi controlului acestora.

- În capitolul 6 am propus utilizarea unor modele matematice pentru estimarea fiabilităţii sistemelor de monitorizare şi control a staţiilor electrice:
  - modelul de distribuţie Rayleigh, prin care am făcut aprecieri asupra eficienţei procesului de testare a aplicaţiilor care achiziţionează date în timp real de la dispozitivele de monitorizare (IED) a echipamentelor electrice;
  - modelul matematic al lanţurilor Markov, prin care se poate estima starea curentă a sistemului şi timpul de trecere dintr-o stare în alta. Se poate prezice momentul când sistemul va cădea;
  - reţelele Bayesiene, pentru calculul probabilităţii de defectare a nodurilor reţelei şi a întregului sistem de monitorizare şi control.

- Capitolul 7 descrie aplicaţia software pe care am dezvoltat-o pentru modelarea matematică a unui sistem de monitorizare şi control, în scopul estimării fiabilităţii sale. Aplicaţia conţine modulul *Parser SCL* şi modulul *Calcul BN*. Modulul *Parser SCL* citeşte informaţiile din fişierul scris în limbajul SCL şi pune la dispoziţia modulului de modelare a reţelei Bayesiene (*Calcul BN*), informaţii reprezentând IED-urile instalate în staţie. Aceste IED-uri monitorizează echipamentele electrice primare şi sunt componente ale sistemului de monitorizare şi control a staţiei electrice. În implementarea modelului matematic al reţelelor Bayesiene din modulul *Calcul BN*, aceste IED-uri sunt reprezentate ca fiind nodurile reţelei Bayesiene. Legăturile între noduri sunt editabile, existând posibilitatea definirii celor două tipuri de noduri: noduri de tip fiu şi noduri de tip părinte.

Configuraţia staţiei electrice în limbajul SCL conform standardului IEC61850 poate fi completată, adăugând noduri de tip server şi noduri pe care rulează aplicaţii software. Utilizând informaţii privind rata de defectare a IED-urilor din studiul de caz (sistemul EMCSIT), am calculat probabilitatea de defectare a serverelor locale, a serverului central precum şi a întregului sistem de monitorizare şi control a staţiei electrice. Scopul acestor calcule este estimarea fiabilităţii sistemului pentru a putea interveni prompt şi eficient în cadrul subsistemelor care prezintă o probabilitate de defectare ridicată.

8.2. Contribuţii proprii

În continuare, menţionez principalele contribuţii originale în domeniul tezei de doctorat, care au fost prezentate în diferite capitole ale tezei:

- Am analizat problemele care pot cauza căderi ale unui sistem de monitorizare şi control a unei staţii electrice şi am propus o serie de soluţii de asigurare a calităţii unui astfel de sistem, care pot contribui la creşterea fiabilităţii sale (capitolele 2 şi 3). Aceste soluţii presupun:
    - Utilizarea unor metode eficiente de testare a aplicaţiilor din componenţa sistemului.

      Analiza prezentată în paragraful 6.1, bazată pe date reale culese în timpul procesului de testare, a evidenţiat importanţa dezvoltării unor simulatoare care să permită testarea înainte de instalarea în mediul real de funcţionare, a aplicaţiilor care primesc date în timp real de la dispozitivele de tip IED, cuplate la diferite tipuri de echipamente electrice. Astfel de teste permit descoperirea unui număr însemnat de defecte în aceste aplicaţii, înainte de testarea întregului sistem de monitorizare în mediul real de funcţionare (instalat în staţie). Efectul financiar poate fi semnificativ: timpul de testare cu întregul sistem de monitorizare instalat în staţia electrică (în funcţiune) este limitat şi din acest motiv şi descoperirea defectelor în această perioadă este redusă.
    - Asigurarea securităţii serverelor, a aplicaţiilor software din componenţa sistemului de monitorizare şi control şi a transmisiei datelor între diversele componente ale sistemului.
    - Implementarea unor tehnici de toleranţă la defecte: folosirea blocurilor de recuperare, tehnici de duplicare, tehnici de reconfigurare şi reîntinerire.

- Pornind de la analiza efectuată în capitolele 2 si 3, am dezvoltat un simulator de IED-uri care poate fi configurat pentru diferite tipuri de echipamente electrice (cap. 4). Acesta permite testarea automată a aplicaţiilor (denumite aplicaţii server în sistemul studiu de caz) care preiau datele transmise în timp real de dispozitivele tip IED incluse într-un sistem de monitorizare şi control a unei staţii electrice. În plus, simulatorul permite efectuarea de teste pentru cazurile în care mai multe IED-uri sunt cuplate la aceeaşi aplicaţie, utilizând protocolul Daisy-Chain. Astfel de teste sunt extrem de dificil de realizat în absenţa unui astfel de simulator, deoarece nu sunt disponibile fizic, în timp util, un număr mare de IED-uri pentru conectarea la aplicaţiile server. Pachetele de date transmise de simulator au fost generate prin metoda *Pairwise testing*, o metodă de testare „black-box" foarte eficientă, care poate conduce la descoperirea a aproximativ 70% din defectele existente într-un software, prin utilizarea unui număr relativ redus de cazuri de test. Deoarece testarea este automată, consumă un timp redus şi nu necesită intervenţia tester-ului pe parcursul testării.

- Am analizat, folosind modelul Rayleigh, eficienţa eliminării defectelor pe parcursul dezvoltării şi testării unor componente dintr-un sistem de monitorizare şi control a unei staţii electrice (paragraful 6.1.). Analiza a evidenţiat importanţa eliminării unui număr cât mai mare de defecte înainte de testarea în condiţii reale de exploatare. Un exemplu de rezultate obţinute în urma calculelor matematice pentru modelul Rayleigh folosit pentru evaluarea procesului de proiectare, dezvoltare şi testare în staţie a aplicaţiilor server din componenţa sistemului studiu de caz este prezentat în anexa 3.

- Am propus modelarea unui sistem de monitorizare şi control printr-un lanţ Markov care presupune existenţa a 4 stări ale sistemului: stare bună, stare acceptabilă, stare proastă şi stare inacceptabilă. S-a propus o metodă de a estima starea curentă a sistemului prin încadrarea într-una din cele 4 stări propuse, utilizând drept variabilă aleatoare „numărul de pachete de date ratate la intervalul fixat de achiziţie de către sistem". Este descrisă metoda de calcul a FPT (first passage times) – timpii de tranziţie între stări – care ajută la predicţia momentului de timp când un sistem poate ajunge într-o stare viitoare. Modelul propus, cu 4 stări ale sistemului (în loc de două, cf. [Ban9-4]) permite predicţia momentului când sistemul poate ajunge într-o stare anterioară celei inacceptabile sau chiar în starea inacceptabilă, astfel încât să se poată interveni cât mai rapid pentru

remedierea defectelor. Implementarea acestui model poate preveni situaţii grave generate de funcţionarea inacceptabilă sau chiar nefuncţionarea sistemului. Propagarea acestor situaţii poate duce chiar la un dezechilibru energetic pe o anumită zonă din reţeaua energetică de transport sau distribuţie ceea ce afectează direct consumatorii finali casnici şi industriali [Urs8-20].

- În scopul estimării probabilităţii de apariţie a defectelor în timpul operării sistemului, am propus utilizarea unei reţele Bayesiene (paragraful 6.3.). Am propus modelarea sistemelor de monitorizare şi control definite conform standardului IEC61850 (anexa 2), printr-o astfel de reţea (paragraful 6.3.2.). De asemenea, am modelat printr-o reţea Bayesiană sistemul de monitorizare şi control prezentat ca studiu de caz în capitolul 4 (paragraful 6.3.3. şi anexa 4) [Urs8-6].

- Am descris configuraţia unei staţii electrice reale, conform standardului IEC61850 (limbajul SCL). Această configuraţie include echipamentele electrice primare din staţie şi IED-urile asociate ce alcătuiesc sistemul de monitorizare şi control a staţiei electrice. În anexa 2, este descrisă configuraţia unei staţii electrice de transformare aparţinând reţelei electrice naţionale de transport a energiei electrice (RET), sub forma unui fişier scris în limbajul SCL.

- Am dezvoltat o aplicaţie software (capitolul 7) care:
  - Preia dintr-un fişier în format SCL configuraţia unui sistem de monitorizare şi control a unei staţii electrice.
  - Permite adăugarea de noi noduri definite de utilizator: noduri de tip calculator Server-local, noduri de tip calculator Client, noduri de tip Server de baze de date, etc.
  - Generează reţeaua Bayesiană prin definirea nodurilor de tipul părinte şi a nodurilor de tip fiu din configuraţia sistemului de monitorizare şi control;
  - Permite adăugarea de informaţii pentru nodurile de tip părinte: numărul de defecte/căderi pe o anumită perioadă de timp sau probabilitatea a priori de defectare a nodului;
  - Estimează probabilitatea de defectare a sistemului;
  - Estimează probabilitatea de defectare (a posteriori) a fiecărui nod al reţelei;

- În paragraful 7.3 sunt redate sintetic rezultatele obţinute prin modelarea printr-o reţea Bayesiană şi estimarea probabilităţii de defectare a sistemului din studiul de caz, folosind aplicaţia. Modelarea şi estimarea detaliată sunt prezentate în anexa 4.

8.3. Planuri de cercetare pentru viitor

În viitor voi încerca să studiez şi să aprofundez următoarele aspecte:

- Utilizarea practică a metodei N-version programming în dezvoltarea ulterioară a unor astfel de sisteme software.
- Simulatorul de IED-uri poate fi dezvoltat în continuare pentru a fi compatibil şi cu alte protocoale de comunicaţie şi tipuri de IED-uri.
- Implementarea posibilităţii de vizualizare grafică a unei reţele Bayesiene.
- Completarea aplicaţiei de calcul a fiabilităţii sistemului cu un modul de calcul a unor metrici software relevante, pentru componentele software ale sistemului.
- Completarea modelului reţelei Bayesiene prin includerea în calcule a unor echipamente electrice care nu sunt monitorizate, deci nu au IED ataşat dar prin defectarea lor mecanică/electrică pot afecta funcţionarea sistemului de monitorizare şi control a staţiei electrice.
- Implementarea modelului matematic pentru predicţia timpilor de defectare bazat pe lanţul Markov, propus în paragraful 6.2;

Lista lucrărilor autorului

Am publicat un articol în buletinul ştiinţific UPB şi un capitol dintr-un volum tipărit:

[Urs8-1] **V. Ursianu**, E. Ursianu, R. Ursianu, "Regression model approach through proper roots", Scientific Bulletin UPB. Series A. Applied Mathematics and Physics Vol. 72/2010, Iss.4/33.

[Urs8-2] **V. Ursianu**, M. Iliescu, "Statistică Aplicată în Inginerie - aspecte teoretice", Ed. Bren.

Am publicat 18 articole la conferinţe naţionale şi internaţionale din care câteva cotate ISI:

[Urs8-3] C. Moldoveanu, V. Brezoianu, A. Vasile, **V. Ursianu**, F. Goni, C. Radu, I. Ioniţa, M. Avramescu, S. Zaharescu, B. Toader: "Intelligent electronic system for continuous monitoring and diagnostic of high voltage substations", CMDM2011 International Conference on Condition Monitoring, Diagnosis and Maintenance, 19-23 septembrie 2011, Bucureşti.

[Urs8-4] C. Moldoveanu, **V. Ursianu**, M. Avramescu, I. Ioniţa, F. Goni, E. Mihalcea, M. Nestor, C. Diaconu, I.D. Haţegan: "Expert systems for condition assessment of power transformers", CMDM2011 International Conference on Condition Monitoring, Diagnosis and Maintenance, 19-23 septembrie 2011, Bucureşti.

[Urs8-5] **V. Ursianu**, Fl. Moldoveanu, R. Ursianu, E. Ursianu: "Software quality assurance for monitoring and control systems in the energy field", CSCS18 International Conference on Control Systems and Computer Science, 24-27 mai 2011, Bucureşti.

[Urs8-6] **V. Ursianu**, R. Ursianu, E. Ursianu: "Bayesian Networks to predict Software Quality", SPSR14 Conferinţa Societăţii de Probabilităţi şi Statistică din România, 29-30 aprilie 2011, Bucureşti.

[Urs8-7] C. Moldoveanu, V. Brezoianu,A. Vasile, **V. Ursianu**, M. Avramescu: "NOVA QA - echipament inteligent pentru măsurarea în clasa A şi monitorizarea on-line a calităţii energiei electrice",Conferinţa Reţele Electrice Inteligente Smart Grids, 2-3 noiembrie 2010, Bran.

[Urs8-8] C. Moldoveanu, V. Brezoianu, A. Vasile, **V. Ursianu**, F. Goni, C. Radu, I. Ioniţă: "Intelligent System for the On-Line Real Time Monitoring of High Voltage Substations" IEEE ISGT2010 Innovative Smart Grid Conference, Goteborg **(cotată ISI).**

[Urs8-9] C. Moldoveanu, V. Brezoianu, A. Vasile, **V. Ursianu**, M. Avramescu:„NOVA QA - echipament inteligent pentru măsurarea în clasa A şi monitorizarea on-line a calităţii energiei electrice", REI Reţele Electrice Inteligente 2010, 21-23 septembrie 2010, Sibiu.

[Urs8-10] C. Moldoveanu, V. Brezoianu, A. Vasile, **V. Ursianu**, E. Mihalcea, F. Goni, C. Radu, I. Ioniţa, S. Zaharescu:„EMCSIT - sisteme inteligente pentru monitorizarea on-line a staţiilor electrice de înaltă tensiune", REI Reţele Electrice Inteligente 2010, 21-23 septembrie 2010, Sibiu.

[Urs8-11] C. Moldoveanu, **V. Ursianu**, V. Brezoianu, A. Vasile, I. Ioniţa, S. Gal, C. Diaconu, V. Zaharescu, T. Fagarasan, M. Oltean, G. Moraru: "Solutions for life management and maintenance optimization for large power transformers", CMD International Conference on Condition Monitoring and Diagnosis 2010, Tokyo **(cotată ISI).**

[Urs8-12] E. Ursianu, R. Ursianu, **V. Ursianu**: "Model Rayleigh generalizat cu aplicaţii", SPSR13 Conferinţa Societăţii de Probabilităţi şi Statistică din România, 16 aprilie 2010, Bucureşti.

[Urs8-13] C. Moldoveanu, V. Brezoianu, **V. Ursianu**, A. Vasile, S. Grigorescu, S. Gal: "EMCSIT System for on-line monitoring of electrical stations: Part I: On-line monitoring of power transformers and shunt reactors", CNEI 2009, 5-6 noiembrie 2009, Buzău.

[Urs8-14] C. Moldoveanu, V. Brezoianu, **V. Ursianu**, A. Vasile, S. Grigorescu, S. Gal:"NOVA-QX - complex power quality monitoring system", CNEI Conferinţa Naţională de Industrie Energetică 2009, 5-6 noiembrie 2009, Buzău.

[Urs8-15] C. Moldoveanu, O. Tutuianu, **V. Ursianu**, A. Vasile: "Abordarea sistemică a gestiunii deşeurilor în sectorul distribuţiei energiei electrice", SNOSE 2009, 28-30 octombrie 2009, Buzău.

[Urs8-16] C. Moldoveanu, **V. Ursianu**, A. Vasile, O. Tutuianu: "Aplicaţie informatica de evidenta şi gestiune a deşeurilor în sectorul distribuţiei energiei electrice", CNEE 2009, 21-23 octombrie 2009, Sinaia.

[Urs8-17] **V. Ursianu**, R. Ursianu, E. Ursianu: "Models for Determining the Quality of an Electrical Equipment" , CSCS17 International Conference on Control Systems and Computer Science 2009, Bucureşti.

[Urs8-18] **V. Ursianu**, R. Ursianu, E. Ursianu: "Mathematical Model for Deterioration Process of an Electrical Station with Multiple Components", MACMESE 2008, WSEAS Conference, Bucureşti **(cotată ISI).**

[Urs8-19] C. Moldoveanu, V. Brezoianu, **V. Ursianu**, A. Vasile: "Sistem pentru monitorizarea parametrilor de calitate ai energiei electrice", International Seminar CIGRE SC 2008, Baia Mare.

[Urs8-20] C. Moldoveanu, R. Ursianu, E. Ursianu, E. Mihalcea, M. Nestor, F. Goni, L. Goia, P. Curiac, **V. Ursianu**: "Determine the optimal moments for investigating technical state of primary equipments for the purpose of assuring the safety levels imputed by the National Company Transelectrica – România", CMD2008 International Conference on Condition Monitoring and Diagnosis, Beijing **(cotată ISI).**

Bibliografie

Calitate software

[Swq9-1] “Software Quality Engineering”, J. Tian,
http://lyle.smu.edu/~tian/SQEbook/slides2/16.pdf

[Swq9-2]”Software System Defect Content Prediction From Development Process And Product Characteristics”, Allen Peter Nikora. PhD Dissertation Thessis.

[Swq9-3] Software Fault Tolerance, Carnegie Mellon University
http://www.ece.cmu.edu/~koopman/des_s99/sw_fault_tolerance/

[Swq9-4] “A survey of software fault tolerance techniques”, Z. Xie, H. Sun, K. Saluja.

[Swq9-5] “Software testing and Quality assurance”, K. Naik, P. Tripathy. Ed. Wiley.

[Swq9-6]”Curs avansat de Ingineria Programelor – Asigurarea calităţii software”, Fl. Moldoveanu.

[Swq9-7] “Methodology of N-version programming”, A. Avizienis, Ed. Wiley.

[Swq9-8] “Metrics and Models în Software Quality Engineering”, S. Khan, Ed. Wiley.

[Swq9-9] “Recovery blocks in action: A system supporting high reliability”, T. Anderson, R. Kerr.

[Swq9-10] “Program result-checking: A theory of testing meets a test of theory”, M. Blum, H. Wasserman.

[Swq9-11] “Designing programs that check their work”, M. Blum, S. Kannan.

[Swq9-12] “Predicting dependability by testing”, D. Hamlet.

[Swq9-13] Curs Sisteme distribuite – Toleranta la defecte, D. Petcu.

[Swq9-14] Standardul SR ISO 8402:1995.

[Swq9-15] „Function Point Software Sizing”, A. DeMarco.

[Swq9-16] Function Point
http://en.wikipedia.org/wiki/Function_point

Smart Grid

[Smg9-1] “NOVA EMCSIT – Sistem complex de monitorizare on-line a unei staţii electrice”, C. Moldoveanu, V. Ursianu, A. Vasile, V. Brezoianu.

[Smg9-2] “Sisteme de timp-real”, T. Letea, Ed. Albastra.

[Smg9-3] GE website

http://gepower.com/prod_serv/products/substation_md/en/monitoring_instr_sys/faraday_ismd.htm
[Smg9-4] ABB website
http://www.abb.com/product/db0003db004281/c12573e700330419c225697300337dbc.aspx
[Smg9-5] AREVA website
http://www.areva-td.com/solutions/liblocal/docs/SBUS_ENT_C20_global.pdf
[Smg9-6] Siemens website
http://www.energy.siemens.com/hq/pool/hq/services/power-transmission-distribution/asset-services/iscm-manager-6seiter-final.pdf
[Smg9-7] Smart Grid wikipedia
http://en.wikipedia.org/wiki/Smart_grid
[Smg9-8] "Contoare inteligente şi Reţele Energetice Inteligente", M. Sanduleac, A. Pop, R. Strutu.
[Smg9-9] The National Institute of Standards and Technology (NIST) Smart Grid Conceptual model
http://smartgrid.ieee.org/ieee-smart-grid/smart-grid-conceptual-model
[Smg9-10] "Arhitecturi de reţele inteligente", D. Federenciuc, I. Silvas, P. Postolache.
[Smg9-11] "Echipamente electrice", L. Popescu.
[Smg9-12] "Este standardul IEC 61850 întotdeauna garantul unor performante ridicate?", S. Gadola, D. Dumitrascu, R. Scarlat.
[Smg9-13] Laborator Instrumentaţie Virtuala, UPB,
http://www.vlab.pub.ro/courses/smart_grids/

Reţele de calculatoare

[Net9-1] IPv6 website
http://ro.wikipedia.org/wiki/IPv6
[Net9-2] ArenaIT
http://www.arenait.net/2007/03/12/tot-ce-trebuie-sa-stiti-despre-ipv6.html
[Net9-3] "Reţele locale de calculatoare. Proiectare şi administrare", A. Munteanu.

IEC61850

[Iec9-1] "Standardizarea web-ului energetic", M. Sanduleac, R. Pop, A. Pop, D. Simhas.
[Iec9-2] Neteon website

http://www.neteon.net/PDFFiles/Goose_Generic-Object-Oriented-Substation-Event.pdf
[Iec9-3] “IEC 61850 Power Industry Communications Standard”, B. Lydon.
[Iec9-4] “Understanding and Simulating the IEC 61850 Standard”, Y. Liang, R. Campbell.
[Iec9-5] “Systems Integration Specialists Company”, IEC61850 evaluation kit CD-ROM Software
http://www.nettedautomation.com/solutions/uca/evalkit/index.html
[Iec9-6] “Understanding and using the IEC61850: a case for meta-modelling“, T. Kostic, O. Preiss, C. Frei.
[Iec9-7] "Real Time Simulation Testing Using IEC 61850", M. Desjardine and others.
[Iec9-8] “Design of IEC61850 based Substation Automation Systems according to Customer Requirements”, K.P. Brand, C. Brunner, W.Wimmer.
[Iec9-9] “IEC 61850 Object Model and Configuration Language”, C. Brunner.
[Iec9-10] “Analysis and Implementation of the IEC 61850 standard”, E. Hammer, E. Sivertsen.
[Iec9-11] Visual SCL, Applied Systems Engineering website
http://www.ase-systems.com/iec-61850/visual-scl.asp

Metrici software

[Swm9-1] “Metrici de complexitate software bazate pe dependentele instrucţiunilor”, I. Ivan, A. Karadimou, A. Licuriceanu, G. Lupu.
[Swm9-2] “Maintainability Index Revisited”, T. Kuipers, J. Visser.
[Swm9-3] Microsoft MSDN website
http://blogs.msdn.com/b/codeanalysis
[Swm9-4] “Software Metrics”, Fl. Moldoveanu.
[Swm9-5]“Compendium of Software Quality Standards and Metrics – Version 1.0”, R. Lincke, W. Lowe.
[Swm9-6] Microsoft MSDN website
http://blogs.msdn.com/b/codeanalysis/archive/2007/10/03/new-for-visual-studio-2008-code-metrics.aspx

Reţele Bayesiene şi modele matematice

[Ban9-1] Webopedia website
http://www.webopedia.com/TERM/S/software_entropy.html
[Ban9-2] Curs SPTR (Sisteme de Programe de Timp Real), Adina Florea.

[Ban9-3] Curs IA (Inteligenta Artificiala), ASE Bucureşti.
[Ban9-4] “Event-based Failure Prediction. An Extended Hidden Markov Model Approach”, F. Salfner. PhD Dissertation Thessis.
[Ban9-5] Math 115, Calculus II with Probability&Metrics, J. Guffin. http://www.math.upenn.edu/~guffin/teaching/spring11/index.html
[Ban9-6] “Predict and approximate software quality with Bayesian Networks and Quality Factors”, H. Motameni, H. Kamfar, A. Khanteimori.
[Ban9-7] “Teoria deciziilor statistică”, V. Preda. Ed. Academiei Romane.
[Ban9-8] "Software Project and Quality Modelling Using Bayesian Networks", N. Fenton, etc.
[Ban9-9] "Predicting Software Defects in Varying Development Lifecycles using Bayesian Nets", N. Fenton, etc.
[Ban9-10] Curs Inteligenta Artificiala, C. Bodea.
[Ban9-11] “Evaluating Software Degradation through Entropy”, A. Bianchi, D. Caivano, F. Lanubile, G. Visaggio.
[Ban9-12] “Dicţionar explicativ de statistică”, V. Clocotici. http://profs.info.uaic.ro/~val/statistica/StatGloss.htm

Fiabilitate

[Rel9-1]“Fiabilitatea în arhitectură calculatoarelor”, M. Budiu.
[Rel9-2] “Dependable computing: From concepts to design diversity“, A. Avizienis, J.-C. Laprie.

Anexe

Anexa 1. Exemplu de configurare a unei staţii electrice în limbajul SCL

Structura tipică a unui fişier scris în limbajul SCL este următoarea:

<informaţii generice XML>

<informaţii generice SCL>

<header>

<history>

Informaţii opţionale privind istoricul staţiei

</history>

</header>

<Substation>

<PowerTransformer>

Informaţii tehnice privind auto/transformatorul de putere

</PowerTransformer>

<VoltageLevel>

<Bay>

Informaţii privind celulele electrice

<ConductingEquipment>

Informaţii tehnice privind fiecare echipament electric primar aparţinând de celula electrică si substaţia aferentă nivelului de tensiune descris

</ConductingEquipment>

<ConductingEquipment>

…

</ConductingEquipment>

</Bay>

<Bay>

Informaţii privind celulele electrice

<ConductingEquipment>

…

</ConductingEquipment>

<ConductingEquipment>

…

</ConductingEquipment>

</Bay>

<LNode>

Informaţii privind alte echipamente electrice primare sau secundare, noduri tip calculator, noduri tip HMI, noduri ce pot fi particularizate în funcţie de situaţie, etc.

</LNode>

...

<LNode>

</LNode>

...

</VoltageLevel>

<IED>Informaţii privind IED-urile asociate nodurilor logice-echipamentelor electrice

</IED>

...

<IED>

</IED>

...

</Substation>

Se utilizează următoarele notaţii:

- Substation = staţia electrică;
- PowerTransformer = auto/transformator de putere;
- VoltageLevel = nivelul de tensiune al substaţiei electrice aparţinând staţiei electrice;
- Bay = celula electrică ce aparţine de substaţia corespunzătoare nivelului de tensiune descris de ‘VoltageLevel’;

ConductingEquipment = Informaţii tehnice privind fiecare echipament electric primar împărţite în următoarele categorii:

- CBR = întreruptor;
- DIS = separator;
- CTR = transformator de măsură de curent;
- VTR = transformator de măsură de tensiune;
- SAR = descărcător;
- RRC = bobina de compensare;

- IED = intelligent electronic device, dispozitivul hardware ce monitorizează şi/sau controlează un echipament electric. Pe acest IED rulează software tip embedded.

Paşii modelării staţiei electrice sunt următorii:

1. Sunt adăugate informaţii privind staţia electrică, nivelul de tensiune;
2. Se completează modelul cu auto/transformatorul de putere, componenta cea mai importantă a staţiei electrice;
3. Pentru fiecare nivel de tensiune - substaţie electrică se vor adăuga echipamentele electrice;
4. Se adaugă IED-urile şi se fac legăturile cu echipamentele electrice;

Staţia electrică în funcţie de nivelul de tensiune de intrare şi nivelul tensiunii de ieşire, este împărţită în două substaţii.

Fiecare substaţie este descrisă în cadrul secţiunii <VoltageLevel>. Ea conţine minim două celule electrice. Fiecare celulă electrică are minim un întreruptor şi două separatoare. În funcţie de specificul staţiei, o celulă electrică mai poate conţine un transformator de măsură de curent, un transformator de măsură de tensiune şi/sau un descărcător.

În cazul celulei electrice pentru transformatorul de mare putere, cu siguranţă aceasta va include minim un descărcător, mai multe separatoare (pentru conexiunile la barele din staţie), transformator de măsură de curent şi transformator de măsură de tensiune.

Chiar dacă se doreşte implementarea unui sistem de monitorizare şi control complet al unei staţii electrice, există posibilitatea ca unele echipamente electrice să nu poată fi monitorizate.

Acest fapt se datorează tipului de echipament electric şi/sau imposibilităţii conectării unor senzori sau traductori pentru achiziţia de semnale de la acesta.

O structură simplă de staţie electrică este reprezentată în limbajul SCL astfel:

```
<?xml version="1.0" encoding="utf-8"?>
<SCL xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.iec.ch/61850/2003/SCL">
<Header toolID="Visual SCL" nameStructure="FuncName">
<History />
</Header>
<Substation name="Staţia Dârste 400/110kV">
<PowerTransformer name="Trafo2">
<TransformerWinding name="W1" />
<TransformerWinding name="W2" />
<TransformerWinding name="W3" />
</PowerTransformer>
<VoltageLevel sxy:x="10" sxy:y="10" name="400kV"
xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates">
<ConductingEquipment name="CBR1" type="CBR" iedName="IED1"/>
<ConductingEquipment name="DIS1" type="DIS" iedName="IED2"/>
<ConductingEquipment name="CTR1" type="CTR" iedName="IED3"/>
<ConductingEquipment name="VTR1" type="VTR" iedName="IED4"/>
<ConductingEquipment name="SAR1" type="SAR" iedName="IED5"/>
</VoltageLevel>
<VoltageLevel sxy:x="406" sxy:y="10" name="110kV"
xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates">
<ConductingEquipment name="CBR2" type="CBR" iedName="IED6"/>
<ConductingEquipment name="DIS2" type="DIS" iedName="IED7"/>
<ConductingEquipment name="CTR2" type="CTR" iedName="IED8"/>
<ConductingEquipment name="VTR2" type="VTR" iedName="IED9"/>
<ConductingEquipment name="SAR2" type="SAR" iedName="IED10"/>
</VoltageLevel>
</Substation>
<IED name="IED1" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED2" desc="" type="" manufacturer="" configVersion="" />
```

<IED name="IED3" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED4" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED5" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED6" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED7" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED8" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED9" desc="" type="" manufacturer="" configVersion="" />
<IED name="IED10" desc="" type="" manufacturer="" configVersion="" />
<LNode lnInst="1" lnClass="IHMI" />
<LNode lnInst="ServerLocal" />
</SCL>

Nodurile de tip IHMI sunt noduri predefinite şi înseamnă Human Machine Interface ce ar fi echivalentul la un terminal (PC) client pe care se afişează parametrii electrici monitorizaţi cu ajutorul IED-urilor. Pe acest IHMI rulează un modul software de tip aplicaţie client. Acest nod poate reprezenta calculatorul din camera de comandă pe care rulează aplicaţia software tip client, de monitorizare şi control a staţiei electrice.

Exista posibilitatea adăugării de noduri definite de utilizator, cum ar fi nodul de tip “ServerLocal” ce intră în categoria User Extension. În această categorie pot intra serverele locale sau serverul central de baze de date.

În varianta complexă, un fişier în format SCL conţine mai multe secţiuni de tip <VoltageLevel> ce corespund substaţiilor electrice. Fiecare astfel de secţiune corespunzătoare unei substaţii va conţine informaţii privind echipamentele electrice şi IED-urile asociate acestora aparţinând doar de aceasta.

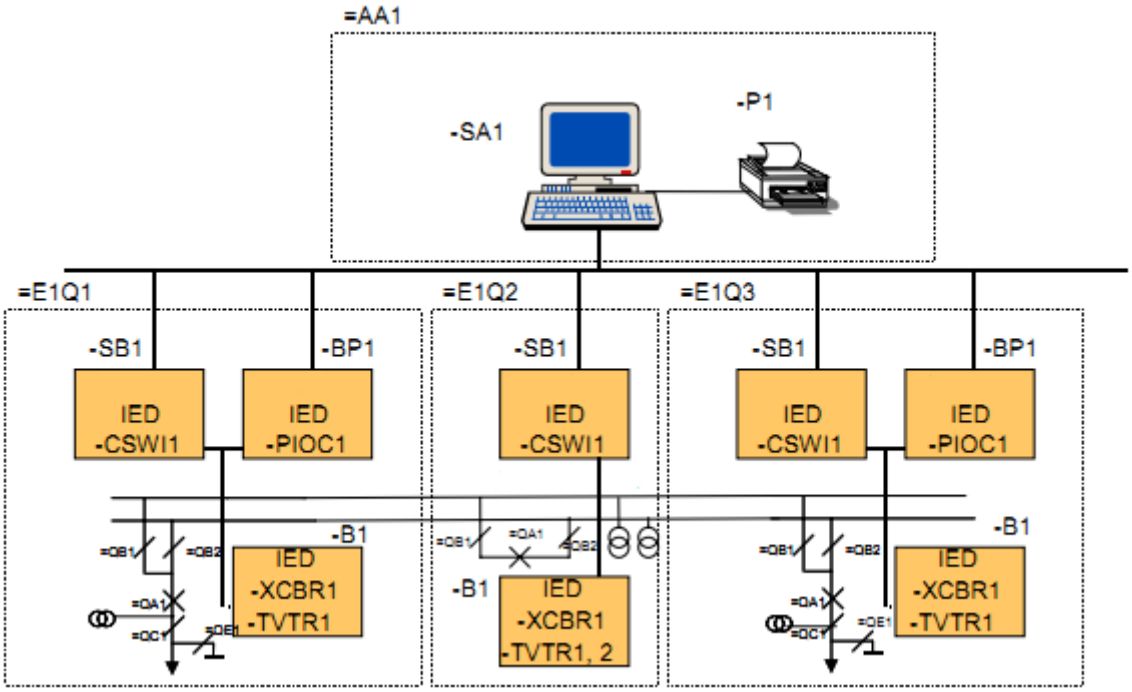

Fig. A1. Exemplu de configurare a unui sistem de monitorizare şi control a unei staţii electrice conform IEC61850-6

Anexa 2. Modelarea în limbajul SCL a staţiei electrice studiu de caz

Fişierul cu extensia .SCD scris în limbajul Substation Configuration Language (SCL) şi care conţine schema staţiei 400/110kV utilizată în cadrul studiului de caz conţine următoarele date:

<?xml version="1.0" encoding="utf-8"?>
<SCL xmlns: xsd="http://www.w3.org/2001/XMLSchema" xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.iec.ch/61850/2003/SCL">
<Header toolID="Visual SCL" nameStructure="FuncName">
<History />
</Header>
<Substation name="Staţia 400/110kV">
<VoltageLevel sxy:x="-54" sxy:y="51" name="Substaţia 400kV" xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates">
<Bay sxy:x="81" sxy:y="107" name="LEA Braşov">
<ConductingEquipment sxy:x="-74" sxy:y="229" name="SB2-bv400" type="DIS" iedName="EMCSIT-S-1-bv400"/>
<ConductingEquipment sxy:x="-75" sxy:y="64" name="SBTF-bv400" type="DIS" iedName="EMCSIT-S-2-bv400"/>
<ConductingEquipment sxy:x="-76" sxy:y="-35" name="D-bv400" type="SAR" iedName="EMCSIT-D-bv400"/>
<ConductingEquipment sxy:x="133" sxy:y="-37" name="TT-bv400" type="VTR" iedName="EMCSIT-TCTT-bv400"/>
<ConductingEquipment sxy:x="19" sxy:y="20" name="Tc-bv400" type="CTR" iedName="EMCSIT-TCTT-bv400"/>
<ConductingEquipment sxy:x="22" sxy:y="116" name="SL-bv400" type="DIS" iedName="EMCSIT-S-3-bv400"/>
<ConductingEquipment sxy:x="22" sxy:y="170" name="I-bv400" type="CBR" iedName="EMCSIT-I-bv400"/>
<ConductingEquipment sxy:x="134" sxy:y="227" name="SB1-bv400" type="DIS" iedName="EMCSIT-S-4-bv400"/>
</Bay>

<VoltageLevel sxy:x="25" sxy:y="8" name="Substaţia 110kV" xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates">
<Bay sxy:x="78" sxy:y="810" name="Zizin2">
<ConductingEquipment sxy:x="6" sxy:y="23" name="SL-z2" type="DIS" iedName="EMCSIT-S-1-z2"/>
<ConductingEquipment sxy:x="-34" sxy:y="175" name="SB-2-z2" type="DIS" iedName="EMCSIT-S-2-z2"/>
<ConductingEquipment sxy:x="54" sxy:y="176" name="SB-1B-z2" type="DIS" iedName="EMCSIT-S-3-z2"/>
<ConductingEquipment sxy:x="76" sxy:y="-6" name="TT-z2" type="VTR" iedName="EMCSIT-TCTT-z2"/>
<ConductingEquipment sxy:x="6" sxy:y="77" name="TC-z2" type="CTR" iedName="EMCSIT-TCTT-z2"/>
<ConductingEquipment sxy:x="6" sxy:y="122" name="I-z2" type="CBR" iedName="EMCSIT-I-z2"/>
</Bay>
<Bay sxy:x="1013" sxy:y="807" name="Sacele2">
<ConductingEquipment sxy:x="125" sxy:y="-3" name="TT-sac2" type="VTR" iedName="EMCSIT-TCTT-sac2"/>
<ConductingEquipment sxy:x="58" sxy:y="28" name="SL-sac2" type="DIS" iedName="EMCSIT-S-1-sac2"/>
<ConductingEquipment sxy:x="58" sxy:y="81" name="TC-sac2" type="CTR" iedName="EMCSIT-TCTT-sac2"/>
<ConductingEquipment sxy:x="58" sxy:y="126" name="I-sac2" type="CBR" iedName="EMCSIT-I-sac2"/>
<ConductingEquipment sxy:x="7" sxy:y="176" name="SB-2-sac2" type="DIS" iedName="EMCSIT-S-2-sac2"/>
<ConductingEquipment sxy:x="119" sxy:y="173" name="SB-1B-sac2" type="DIS" iedName="EMCSIT-S-3-sac2"/>
</Bay>
<Bay sxy:x="1780" sxy:y="801" name="Trafo2-110">
<ConductingEquipment sxy:x="36" sxy:y="8" name="TT-T2-110" type="VTR" iedName="EMCSIT-TCTT-T2-110"/>

<ConductingEquipment sxy:x="12" sxy:y="92" name="TC-T2" type="CTR" iedName="EMCSIT-TCTT-T2-110"/>
<ConductingEquipment sxy:x="11" sxy:y="134" name="I-T2-110" type="CBR" iedName="EMCSIT-I-T2-110"/>
<ConductingEquipment sxy:x="-30" sxy:y="179" name="SB-2-T2" type="DIS" iedName="EMCSIT-S-1-T2"/>
<ConductingEquipment sxy:x="47" sxy:y="182" name="SB-1A-T2" type="DIS" iedName="EMCSIT-S-2-T2"/>
</Bay>
</VoltageLevel>
</Substation>
<IED name="EMCSIT-S-1-bv400" />
<IED name="EMCSIT-S-2-bv400"/>
<IED name="EMCSIT-D-bv400"/>
<IED name="EMCSIT-TCTT-bv400"/>
<IED name="EMCSIT-TCTT-bv400"/>
<IED name="EMCSIT-S-3-bv400"/>
<IED name="EMCSIT-I-bv400"/>
<IED name="EMCSIT-S-4-bv400"/>

<IED name="EMCSIT-TCTT-z2" />
<IED name="EMCSIT-S-1-z2" />
<IED name="EMCSIT-I-z2" />
<IED name="EMCSIT-S-2-z2" />
<IED name="EMCSIT-S-3-z2" />

<IED name="EMCSIT-TCTT-sac2" />
<IED name="EMCSIT-S-1-sac2" />
<IED name="EMCSIT-I-sac2" />
<IED name="EMCSIT-S-2-sac2" />
<IED name="EMCSIT-S-3-sac2" />

<IED name="EMCSIT-TCTT-T2-110" />
<IED name="EMCSIT-S-1-T2-110" />

<IED name="EMCSIT-I-T2-110" />
<IED name="EMCSIT-S-2-T2-110" />
<IED name="EMCSIT-D-T2" />

În selecţia prezentată sunt enumerate echipamentele electrice din cadrul staţiei electrice, grupate în 4 celule electrice.

Prima celulă aparţine de substaţia de 400kV, corespunzătoare secţiunii <VoltageLevel sxy:x="-54" sxy:y="51" name="Substaţia 400kV" xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates">.

Următoarele trei celule aparţin de substaţia de 110kV, corespunzătoare secţiunii <VoltageLevel sxy:x="25" sxy:y="8" name="Substaţia 110kV" xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates">.

Aceste celule sunt grupate în secţiuni delimitate de separatorul <Bay>

În partea finală a fişierului sunt enumerate IED-urile ce monitorizează echipamentele electrice şi care alcătuiesc sistemul de monitorizare şi control.

Legătura IED-urilor cu echipamentele electrice este făcută în cadrul secţiunii ce descrie echipamentele electrice: <ConductingEquipment> prin completarea atributului „iedName” cu numele IED-ului conectat.

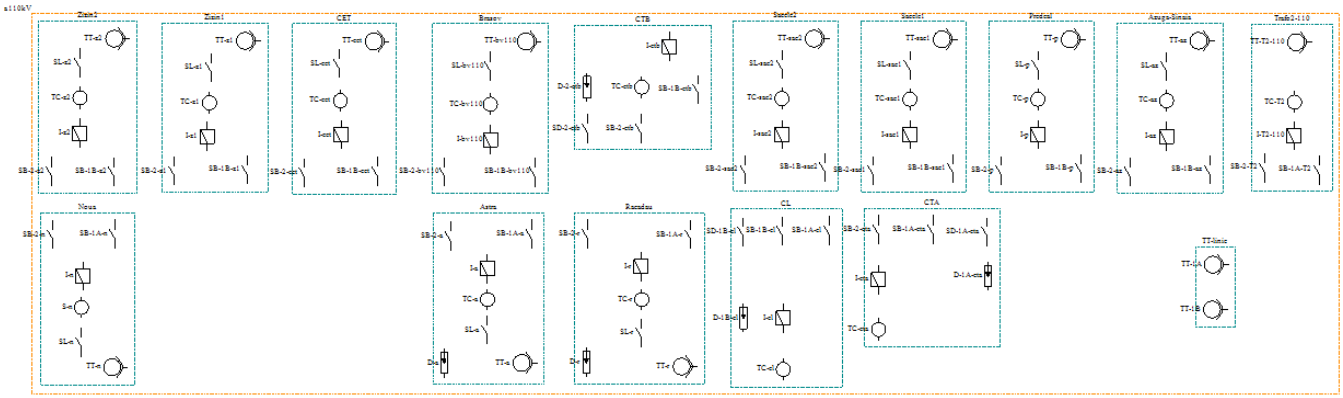

Fig. A2.1. Exemplu de modelare grafică a substaţiei de 110kV studiu de caz

Datele reprezintă o parte din configuraţia realizată pentru staţia electrică din studiul de caz.

Anexa 3. Rezultatele matematice pentru modelul de distribuţie Rayleigh

Definirea modelului Rayleigh:

Y = x/(σ * σ)*e^(-(x*x)/(2* σ * σ))

unde σ este parametrul de scală al modelului Rayleigh şi x este variabila aleatoare.

Numărul de observaţii = 11

Rezultatele aplicării modelului Rayleigh pentru ratele de defectare corespunzătoare aplicaţiei server pentru IED1

În urma aplicării modelului Rayleigh pe datele de intrare reprezentate de numărul de defecte din perioada de testare a aplicaţiei server pentru IED1, a rezultat că se potriveşte cel mai bine curba Rayleigh cu parametrul σ = 3,9894.

Tabel de valori obţinute prin aplicarea modelului Rayleigh

Număr luni	f(x) funcţia de densitate de probabilitate
1	0,060889291
2	0,11082536
3	0,142072758
4	0,152034487
5	0,143237539
6	0,121662008
7	0,094347591
8	0,067307579
9	0,044388438
10	0,027151441
11	0,015440567

Rezultatele aplicării modelului Rayleigh pentru ratele de defectare corespunzătoare aplicaţiei server pentru IED2

În urma aplicării modelului Rayleigh pe datele de intrare reprezentate de numărul de defecte din perioada de testare a aplicaţiei server pentru IED2, a rezultat că se potriveşte cel mai bine curba Rayleigh cu parametrul σ = 4,5697

Tabel de valori obţinute prin aplicarea modelului Rayleigh

Număr luni	f(x) funcţia de densitate de probabilitate
1	0,04675477
2	0,087028165
3	0,115813077
4	0,130588657
5	0,131591305
6	0,121345247
7	0,103701847
8	0,082755712
9	0,061968738
10	0,043687268
11	0,029065308

Rezultatele aplicării modelului Rayleigh pentru ratele de defectare corespunzătoare aplicaţiei server pentru IED3

În urma aplicării modelului Rayleigh pe datele de intrare reprezentate de numărul de defecte din perioada de testare a aplicaţiei server pentru IED3, a rezultat că se potriveşte cel mai bine curba Rayleigh cu parametrul $\sigma = 3,4075$

Tabel de valori obţinute prin aplicarea modelului Rayleigh

Număr luni	f(x) funcţia de densitate de probabilitate
1	0,082494787
2	0,144994393
3	0,175361255
4	0,172965716
5	0,146741907
6	0,109651858
7	0,073086951
8	0,04378298
9	0,02368797
10	0,011613197

11	0,005171392

Rezultatele aplicării modelului Rayleigh pentru ratele de defectare corespunzătoare aplicaţiei server pentru IED4

În urma aplicării modelului Rayleigh pe datele de intrare reprezentate de numărul de defecte din perioada de testare a aplicaţiei server pentru IED4, a rezultat că se potriveşte cel mai bine curba Rayleigh cu parametrul $\sigma = 3,4817$

Tabel de valori obţinute prin aplicarea modelului Rayleigh

Număr luni	f(x) funcţia de densitate de probabilitate
1	0,079159705
2	0,139892605
3	0,170734126
4	0,170556047
5	0,147081792
6	0,112123246
7	0,076519423
8	0,047104955
9	0,026284263
10	0,013338399
11	0,006170503

Rezultatele aplicării modelului Rayleigh pentru ratele de defectare corespunzătoare aplicaţiei server pentru IED5

În urma aplicării modelului Rayleigh pe datele de intrare reprezentate de numărul de defecte din perioada de testare a aplicaţiei server pentru IED5, a rezultat că se potriveşte cel mai bine curba Rayleigh cu parametrul $\sigma = 2,6458$

Tabel de valori obţinute prin aplicarea modelului Rayleigh

Număr luni	f(x) funcţia de densitate de probabilitate
1	0,133004423
2	0,214702154
3	0,225334763

4	0,182233276
5	0,119772927
6	0,065512035
7	0,030200162
8	0,011822323
9	0,003949273
10	0,001129527
11	0,00027725

Anexa 4. Estimarea probabilităţii de defectare a sistemului de monitorizare şi control utilizând reţeaua Bayesiană

Se calculează probabilitatea de defectare a subsistemului de monitorizare şi control aferent cabinei de relee 1: **Server local CR1.**

În total sunt 11 IED-uri în cabina de relee 1: 2 IED-uri de tip IED1 (A1), 5 IED-uri de tip IED2 (A2), 2 IED-uri de tip IED3 (A3) şi 2 IED-uri de tip IED4 (A4) .Utilizând aceste informaţii, se calculează probabilităţile pentru nodurile de tip IED din cabina de relee 1:

P(A1) = 2/11 = 0,18

P(A2) = 5/11 = 0,46

P(A3) = 2/11 = 0,18

P(A4) = 2/11 = 0,18

Utilizând informaţiile din tabelul A4.1 se calculează probabilităţile de defectare a priori ale IED-urilor din cabina de relee 1.

Tabel A4.1. Număr de căderi pentru fiecare IED, în perioada de 4 luni de test în staţia electrică

IED1	**IED2**	**IED3**	**IED4**
2 căderi / 2 IED-uri diferite	1 cădere / 1 IED	0 căderi	0 căderi

Pentru IED-urile de tip IED1 din cabina de relee 1, au fost constatate 2 căderi la 2 IED-uri diferite de tip IED1, în perioada de 4 luni.

P(E=defect|A1) = 0,25

Pentru IED-urile de tip IED2 din cabina de relee 1, a fost constatată 1 cădere la 1 singur IED din totalul de 5 IED-uri de tip IED2, în perioada de 4 luni.

P(E=defect|A2) = 0,05

Pentru IED-urile de tip IED3 din cabina de relee 1, n-a fost constatată nici o cădere în perioada de 4 luni.

P(E=defect|A3) = 0

Pentru IED-urile de tip IED4 din cabina de relee 1, n-a fost constatată nici o cădere în perioada de 4 luni.

P(E=defect|A4) = 0

Se calculează probabilitatea de defectare a nodului tip server local din cabina de relee 1, notat în continuare CR1, modelat prin reţeaua Bayesiană:

P(CR1) = P(A1) x P(E=defect|A1) + P(A2) x P(E=defect|A2) + P(A3) x P(E=defect|A3) + P(A4) x P(E=defect|A4) = 0,25 x 0,18 + 0,05 x 0,46 = 0,068

Deci, probabilitatea de defectare a subsistemului de monitorizare din cabina de relee 1 este de 6,8%.

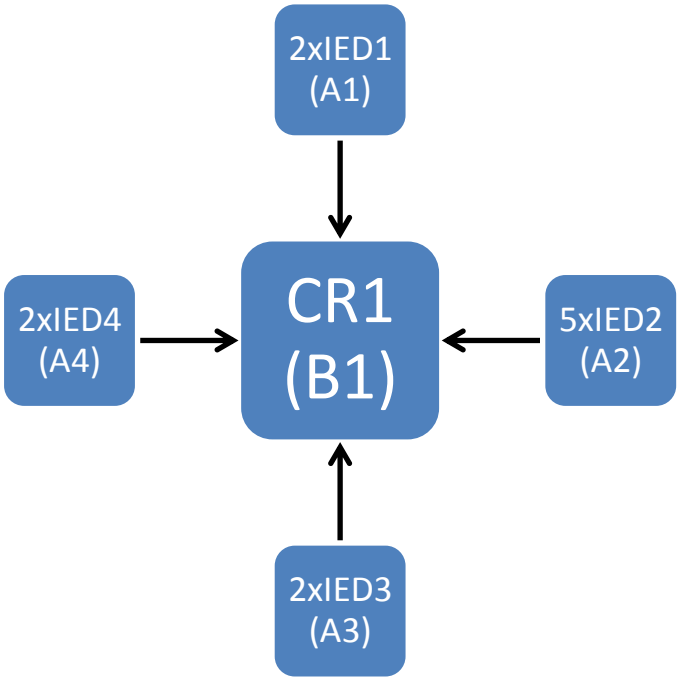

Fig. A3.1. Distribuţia IED-urilor pentru cabina de relee CR1 şi notaţia conform reţelei Bayesiene

Se calculează probabilitatea de defectare a subsistemului de monitorizare şi control aferent cabinei de relee 2: **Server local CR2.**

În total sunt 16 IED-uri în cabina de relee 2: 3 IED-uri de tip IED1 (A1), 9 IED-uri de tip IED2 (A2), 1 IED de tip IED3 (A3) şi 3 IED-uri de tip IED4 (A4) .Utilizând aceste informaţii, se calculează probabilităţile pentru nodurile de tip IED din cabina de relee 2:

P(A1) = 3/16 = 0,1875

P(A2) = 9/16 = 0,5625

P(A3) = 1/16 = 0,0625

P(A4) = 3/16 = 0,1875

Utilizând informaţiile din următorul tabel:

Tabel A4.2. Număr de căderi pentru fiecare IED, în perioada de 4 luni de test în staţia electrică

IED1	**IED2**	**IED3**	**IED4**
0 căderi	3 căderi / 3 IED-uri	1 cădere / 1 IED	0 căderi

se calculează probabilităţile de defectare a priori ale IED-urilor din cabina de relee 2.

Pentru IED-urile de tip IED1 din cabina de relee 2, n-a fost constatată nici o cădere în perioada de 4 luni.

P(E=defect|A1) = 0

Pentru IED-urile de tip IED2 din cabina de relee 2, au fost constatate 3 căderi la 3 IED-uri diferite din totalul de 9 IED-uri de tip IED2, în perioada de 4 luni.

P(E=defect|A2) = 0,083

Pentru IED-urile de tip IED3 din cabina de relee 2, a fost constatată 1 cădere la 1 IED din totalul de 3 IED-uri de tip IED3, în perioada de 4 luni.

P(E=defect|A3) = 0,083

Pentru IED-urile de tip IED4 din cabina de relee 2, n-a fost constatată nici o cădere în perioada de 4 luni.

P(E=defect|A4) = 0

Se calculează probabilitatea de defectare a nodului tip server local din cabina de relee 2, notat în continuare CR2, modelat prin reţeaua Bayesiană:

P(CR1|E=defect) = P(A1) x P(E=defect|A1) + P(A2) x P(E=defect|A2) + P(A3) x P(E=defect|A3) + P(A4) x P(E=defect|A4) = 0,083 x 0,5625 + 0,083 x 0,1875 = 0,052

Deci, probabilitatea de defectare a subsistemului de monitorizare din cabina de relee 2 este de 5,2%.

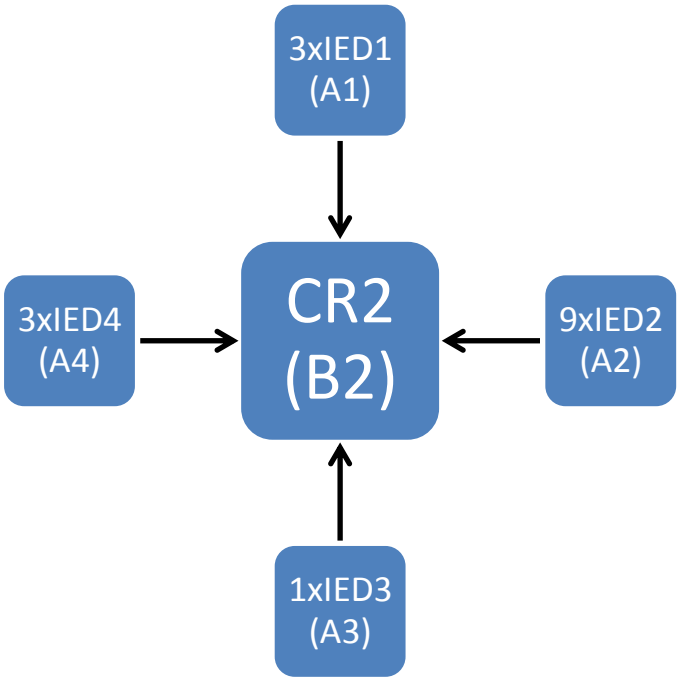

Fig. A3.2. Distribuţia IED-urilor pentru cabina de relee CR2 şi notaţia conform reţelei Bayesiene

Se calculează probabilitatea de defectare a subsistemului de monitorizare şi control aferent cabinei de relee 3: **Server local CR3.**

În total sunt 23 IED-uri în cabina de relee 3: 4 IED-uri de tip IED1 (A1), 12 IED-uri de tip IED2 (A2), 3 IED-uri de tip IED3 (A3) şi 4 IED-uri de tip IED4 (A4) .Utilizând aceste informaţii, se calculează probabilităţile pentru nodurile de tip IED din cabina de relee 3:

P(A1) = 4/23 = 0,17

P(A2) = 12/23 = 0,52

P(A3) = 3/23 = 0,13

P(A4) = 4/23 = 0,18

Utilizând informaţiile din tabelul A4.3 se calculează probabilităţile de defectare a priori ale IED-urilor din cabina de relee 3.

Tabel A4.3. Număr de căderi pentru fiecare IED, în perioada de 4 luni de test în staţia electrică

IED1	**IED2**	**IED3**	**IED4**
4 căderi / 4 IED-uri	12 căderi / 12 IED-uri	3 căderi / 3 IED-uri	4 căderi / 4 IED-uri

Pentru IED-urile de tip IED1 din cabina de relee 3, au fost constatate 4 căderi la 4 IED-uri diferite din totalul de 4 IED-uri de tip IED1, în perioada de 4 luni.

P(E=defect|A1) = 0,25

Pentru IED-urile de tip IED2 din cabina de relee 3, au fost constatate 12 căderi la 12 IED-uri diferite din totalul de 12 IED-uri de tip IED2, în perioada de 4 luni.

P(E=defect|A2) = 0,25

Pentru IED-urile de tip IED3 din cabina de relee 3, au fost constatate 3 căderi la 3 IED-uri din totalul de 3 IED-uri de tip IED3, în perioada de 4 luni.

P(E=defect|A3) = 0,33

Pentru IED-urile de tip IED4 din cabina de relee 3, au fost constatate 4 căderi la 4 IED-uri diferite din totalul de 4 IED-uri de tip IED4, în perioada de 4 luni.

P(E=defect|A4) = 0,25

Se calculează probabilitatea de defectare a nodului tip server local din cabina de relee 3, notat în continuare CR3, modelat prin reţeaua Bayesiană:

P(CR3|E=defect) = P(A1) x P(E=defect|A1) + P(A2) x P(E=defect|A2) + P(A3) x P(E=defect|A3) + P(A4) x P(E=defect|A4) = 0,25 x 0,17 + 0,25 x 0,52 + 0,33 x 0,13 + 0,25 x 0,18 = 0,26

Deci, probabilitatea de defectare a subsistemului de monitorizare din cabina de relee 3 este de 26%.

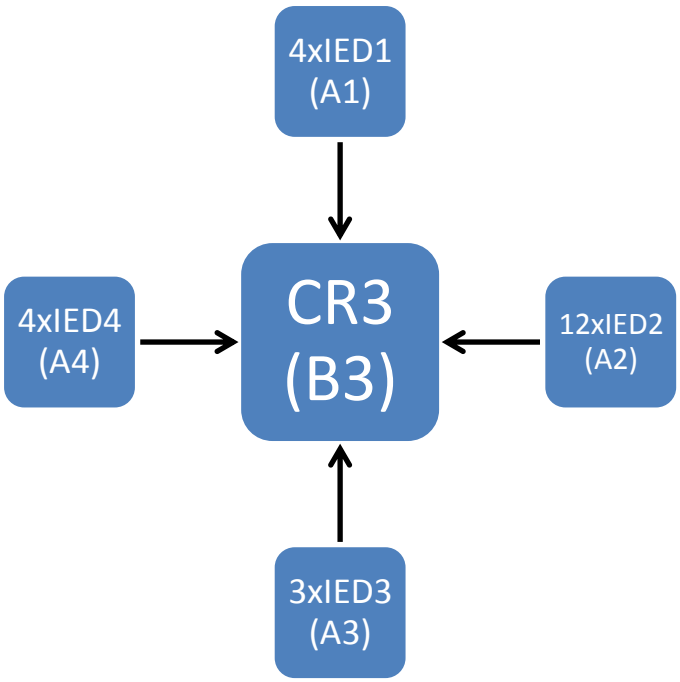

Fig. A3.3. Distribuţia IED-urilor pentru cabina de relee CR3 şi notaţia conform reţelei Bayesiene

Se calculează probabilitatea de defectare a subsistemului de monitorizare şi control aferent cabinei de relee 4: **Server local CR4.**

În total sunt 16 IED-uri în cabina de relee 4: 3 IED-uri de tip IED1 (A1), 9 IED-uri de tip IED2 (A2), 1 IED de tip IED3 (A3) si 3 IED-uri de tip IED4 (A4) .Utilizând aceste informaţii, se calculează probabilităţile pentru nodurile de tip IED din cabina de relee 4:

P(A1) = 3/16 = 0,1875

P(A2) = 9/16 = 0,5625

P(A3) = 1/16 = 0,0625

P(A4) = 3/16 = 0,1875

Utilizând informaţiile din tabelul A4.4 se calculează probabilităţile de defectare a priori ale IED-urilor din cabina de relee 4.

Tabel A4.4. Număr de căderi pentru fiecare IED, în perioada de 4 luni de test în staţia electrică

IED1	**IED2**	**IED3**	**IED4**
0 căderi	0 căderi	0 căderi	0 căderi

Pentru IED-urile de tip IED1 din cabina de relee 4, nu au fost înregistrate căderi.

P(E=defect|A1) = 0

Pentru IED-urile de tip IED2 din cabina de relee 4, nu au fost înregistrate căderi.

P(E=defect|A2) = 0

Pentru IED-urile de tip IED3 din cabina de relee 4, nu au fost înregistrate căderi.

P(E=defect|A3) = 0

Pentru IED-urile de tip IED4 din cabina de relee 4, nu au fost înregistrate căderi.

P(E=defect|A4) = 0

Se poate calcula probabilitatea de defectare a nodului tip server local din cabina de relee 4, notat în continuare CR4, modelat prin reţeaua Bayesiană:

P(CR4|E=defect) = P(A1) x P(E=defect|A1) + P(A2) x P(E=defect|A2) + P(A3) x P(E=defect|A3) + P(A4) x P(E=defect|A4) = 0

Deci, probabilitatea de defectare a subsistemului de monitorizare din cabina de relee 4 este de 0%.

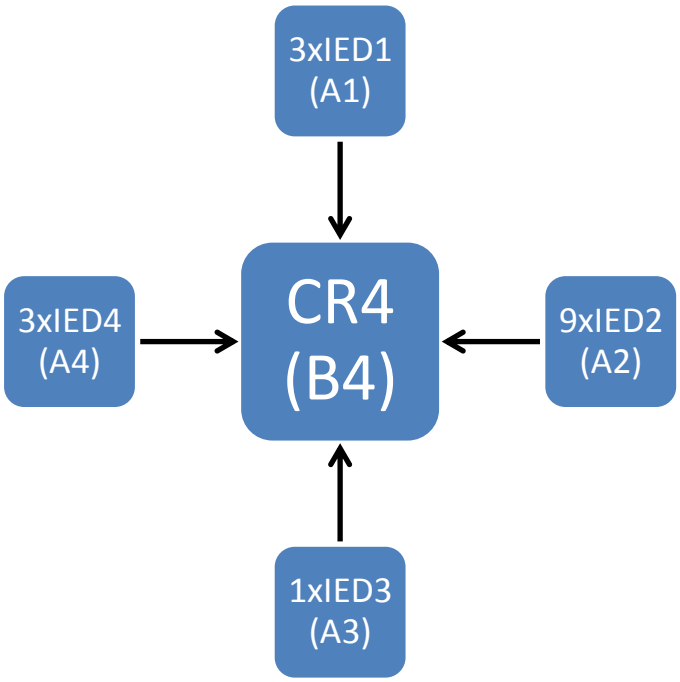

Fig. A3.4. Distribuţia IED-urilor pentru cabina de relee CR4 şi notaţia conform reţelei Bayesiene

Se calculează probabilitatea de defectare a subsistemului de monitorizare şi control aferent cabinei de relee 5: **Server local CR5.**

În total sunt 15 IED-uri în cabina de relee 5: 3 IED-uri de tip IED1 (A1), 9 IED-uri de tip IED2 (A2), nici un IED de tip IED3 (A3) si 3 IED-uri de tip IED4 (A4) .Utilizând aceste informaţii, se calculează probabilităţile pentru nodurile de tip IED din cabina de relee 5:

P(A1) = 3/15 = 0,2

P(A2) = 9/15 = 0,6

P(A3) = 0/15 = 0

P(A4) = 3/15 = 0,2

Utilizând informaţiile din tabelul A4.5, se calculează probabilităţile de defectare a priori ale IED-urilor din cabina de relee 5.

Tabel A4.5. Număr de căderi pentru fiecare IED, în perioada de 4 luni de test în staţia electrică

IED1	**IED2**	**IED4**
0 căderi	3 căderi / 3 IED-uri	3 căderi / 3 IED-uri

Pentru IED-urile de tip IED1 din cabina de relee 5, nu au fost constatate căderi la cele 3 IED-uri diferite, în perioada de 4 luni.

P(E=defect|A1) = 0

Pentru IED-urile de tip IED2 din cabina de relee 5, au fost constatate 3 căderi la 3 IED-uri diferite din totalul de 9 IED-uri de tip IED2, în perioada de 4 luni.

P(E=defect|A2) = 0,083

Nu există IED-uri de tip IED3 în cabina de relee 5.

P(E=defect|A3) = 0

Pentru IED-urile de tip IED4 din cabina de relee 5, a fost constatată o cădere la 1 IED din totalul de 3 IED-uri de tip IED4, în perioada de 4 luni.

P(E=defect|A4) = 0,083

Se calculează probabilitatea de defectare a nodului tip server local din cabina de relee 5, notat în continuare CR5, modelat prin reţeaua Bayesiană:

P(CR5|E=defect) = P(A1) x P(E=defect|A1) + P(A2) x P(E=defect|A2) + P(A3) x P(E=defect|A3) + P(A4) x P(E=defect|A4) = 0,083 x 0,6 + 0,083 x 0,2 = 0,198 + 0,066 = 0,066

Deci, probabilitatea de defectare a subsistemului de monitorizare din cabina de relee 5 este de 6,6%.

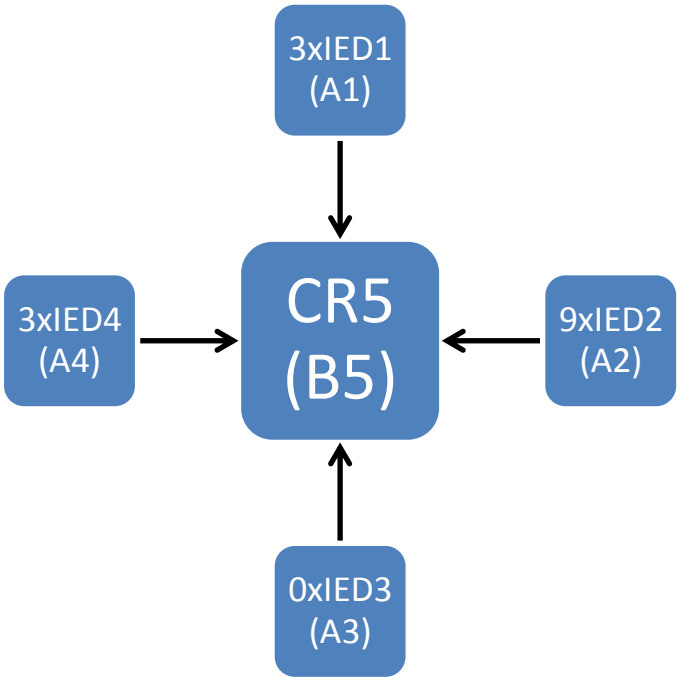

Fig. A3.5. Distribuţia IED-urilor pentru cabina de relee CR5 şi notaţia conform reţelei Bayesiene

Anexa 5. Exemplu de date de test generate prin metoda Pairwise testing

Se iau în considerare 6 mărimi analogice ce pot lua fiecare câte 6 valori:

Parametrul 0 (mărime analogica MA1): -1, 0, 1, 399, 400, 401

Parametrul 1 (mărime analogica MA2): -1, 0, 1, 399, 400, 401

Parametrul 2 (mărime analogica MA3): -1, 0, 1, 399, 400, 401

Parametrul 3 (mărime analogica MA4): -1, 0, 1, 299, 300, 301

Parametrul 4 (mărime analogica MA5): -1, 0, 1, 299, 300, 301

Parametrul 5 (mărime analogica MA6): -1, 0, 1, 299, 300, 301

O selecţie a cazurilor de test obţinute utilizând metoda *Pairwise testing* este prezentată în continuare:

Cazul de test 0: -1 -1 -1 -1 -1 -1

1: 0 0 0 -1 0 0

2: 1 1 -1 0 1 0

3: 399 399 -1 1 0 1

4: 400 400 1 -1 1 1

5: 401 401 0 0 -1 1

6: 1 -1 399 299 0 299

7: 399 -1 400 300 299 0

8: -1 399 401 301 300 0

9: 0 400 -1 299 301 300

10: 400 0 399 0 299 301

11: 401 1 1 300 0 -1

12: 0 401 400 1 1 -1

13: 399 1 401 -1 301 299

14: 1 0 1 301 -1 300

15: 1 401 -1 300 300 301

16: -1 400 0 1 299 299

17: 400 399 0 299 301 -1

18: 401 399 399 -1 1 300

19: 401 400 400 301 0 301

20: 0 -1 401 301 299 1

21: 399 400 399 0 300 -1

22: -1 0 400 299 300 1

23: 400 1 400 1 -1 300

24: 399 401 1 299 299 0

25: 0 399 401 300 -1 299

26: -1 -1 1 1 301 301

27: 400 401 401 0 0 300

28: 401 0 -1 1 300 299

29: 1 -1 0 -1 300 300

30: -1 1 399 300 301 1

31: 399 1 0 301 1 301

32: 1 399 400 0 301 299

33: 0 1 1 0 300 299

34: 401 -1 401 299 1 301

35: 400 400 399 300 -1 0

36: 400 401 399 301 301 299

37: 1 0 401 1 299 -1

38: -1 401 -1 -1 299 300

39: 1 400 401 299 -1 1

40: 399 0 0 300 1 300

41: 400 -1 -1 301 300 -1

42: 0 0 399 1 301 0

43: -1 -1 -1 0 0 -1

44: -1 -1 400 -1 1 299

45: 399 399 1 -1 -1 301

46: 401 1 -1 299 299 0

47: 401 -1 -1 -1 301 -1

48: 0 -1 -1 -1 -1 301

49: -1 399 -1 -1 299 -1